مدخل إلى علم الحاسوب والبرمجة بلغة باسكال

د. غازي إبراهيم رحو

أستاذ مشارك

د. إبراهيم عبد الواحد نائب

أستاذ مشارك

د. محمّد نوفل ضاهر

أستاذ مساعد

قسم الحاسوب وأنظمة المعلومات

جامعة العلوم التطبيقية



دار المناهج للنشر والتوزيع

مدخل إلى علم الحاسوب
والبرمجة بلغة باسكال





دار المناهج للنشر والتوزيع

هاتف ٤٦٥٠٦٢٤ - فاكس ٤٦٥٠٦٢٤ (٦ ٩٦٢)

ص.ب ٢١٥٣٠٨ عمّان ١١١٢٢ الأردن

الطبعة الأولى

١٤٢٠هـ - ١٩٩٩م

رقم الايداع لدى دائرة المكتبات والوثائق الوطنية
(١١٩٠ / ٧ / ١٩٩٩)
رقم الإجازة المتسلسل لدى دائرة المطبوعات والنشر
(٧٨٦ / ٧ / ١٩٩٩)

الإهداء

إلى من صبروا ... ووقفوا إلى جانبنا طيلة

مراحل اعداد هذا الكتاب

زوجاتنا ... أطفالنا

مع خالص الحب والتقدير عرفانا منا لهم

المؤلفون

شكر

يتقدم المؤلفون بجزيل الشكر والعرفان لكل من تقدم باقتراحات وإرشادات وملاحظات حول طباعة وإخراج هذا الكتاب ، وخاصة الأساتذة الأفاضل في قسم الحاسوب وأنظمة المعلومات – جامعة العلوم التطبيقية .

لهم منا جزيل الشكر على ملاحظاتهم القيمة، جزاهم الله خيرا

المؤلفون

المحتويات

الفصل الثالث
الانظمة العددية

الفصل الرابع
تمثيل البيانات والتعليمات داخل الحاسوب

الفصل الخامس
الكيان المنطقي للحاسوب (البرمجيات)

الفصل السادس

الشبكات الحاسوبية وتراسل المعلومات

الفصل السابع
لغات البرمجة

الفصل الثامن
برمجة الحاسوب والمخططات الانسيابية

الفصل التاسع
البرمجة بلغة باسكال
(أساسيات ومكونات – جمل الإدخال والإخراج – جمل التحكم والتكرار)

الفصل العاشر
البرمجة بلغة باسكال

(بيانات المستخدم التعريفية-المصفوفات-البرامج الفرعية-السجلات والملفات)

الفصل الحادي عشر

أساسيات بيئة باسكال

مُقَدِّمَة

نعيش اليوم عصراً يسمى بعصر المعلوماتية، ينصب فيه الاهتمام على جمع كم من المعلومات لتخزينها واسترجاعها وقت الحاجة بهدف الاستفادة منها لاستخلاص النتائج المفيدة في عمليات اتخاذ القرارات الرشيدة والمدروسة في شتى مجالات الحياة.

لقد فرض الحاسوب نفسه كأداة هذا العصر طوعاً أو كرهاً في شتى ميادين المعرفة، أداة لا غنى عنها للباحثين، وطلاب المعرفة، للسياسيين، والاقتصاديين، والإداريين، وللأطباء، والمهندسين ... في المؤسسة، والمصنع، والبيت، والمكتب، والعيادة ...

لقد اتجه العلماء في مناطق عديدة من العالم إلى تطوير الحاسوب وتسهيل التعامل معه و استخدامه، حيث بدأ حياته في النصف الثاني من هذا القرن ضخماً ثقيلاً لا يعمل إلا ضمن ظروف معينة، والتعامل معه كان صعباً و يحتاج إلى تدريب وخبرة طويلة.

لم يمض على عمره أكثر من خمسين عاما وإذ به يغدو خفيفا صغيرا سهل التعامل معه وبمتناول جميع الناس على مختلف مستوياتهم الاجتماعية والعلمية، وحتى الأطفال.

لقد كان الهدف من اختراعه آنذاك هو القيام بأعمال حسابية بسيطة، ليعطيك ما تريده منها في زمن قصير، يعالج البيانات مهما كان حجمها وبسرعة فائقة، يساعد في التصميم الهندسي وفي تشخيص الكثير من الأمراض، كما أنه يمد يد العون عند اتخاذ القرار، وأصبح أيضاً قادراً أن يحل محل العامل في المصنع و المتجر، ناهيك عن دوره في غزو الفضاء وتطوير الأسلحة والتواصل العلمي بين الباحثين ... وتراسل المعلومات عبر شبكات، تدعى الشبكات الحاسوبيه، تتصل مع شتى بقاع العالم.

لذا رأينا من واجبنا تعريف الطالب والقارئ بأهمية الحاسوب وإعطاءه الأسس والمبادئ والخصائص التي يجب أن يلم بها بغية مواكبة عصر المعلوماتية وتمكنه من تسخير إمكانيات الحاسوب الهائلة في خدمته أياً كان موقعه. وقد راعينا عند تصنيفنا للموضوعات التدرج العلمي في الشرح والعرض المبسط، بحيث يحقق للقارئ ما يجب أن يلم به للاستفادة من معلومات هذا الكتاب.

يتضمن هذا الكتاب أحد عشر فصلاً، نقدم في الفصل الأول شرحاً عن أهمية الحاسوب و تطوره وأنواعه، وفي الفصل الثاني نتعرض إلى مكونات الحاسوب المادية (الذاكرة، و المعالج، و وحدات التخزين الرئيسية، والثانوية، وحدات الإدخال والإخراج)، أما الفصل الثالث فيتضمن الأنظمة العددية التي يعتمد عليها الحاسوب في معالجة البيانات ثم بيّنا في الفصل الرابع كيفية تمثيل البيانات حاسوبياً من أعداد ورموز وتعليمات أما في الفصل الخامس فقد تناولنا موضوع الكيان المنطقي للحاسوب وبرمجياته المختلفة وأنظمة التشغيل، وفي الفصل السادس أعطينا فكرة عن الشبكات الحاسوبية وشبكات الأنترنت وتراسل المعلومات، وفي الفصل السابع تناولنا لغات البرمجة وأنواعها، أما في

الفصل الثامن فقد حرصنا على تعليم الطالب و القارئ كيفية معالجة المسائل والمشاكل حاسوبياً من خلال مراحل متسلسلة مروراً بالتحليل المنطقي للمشكلة ثم رسم مخطط سير العمليات، ثم استخدام إحدى لغات البرمجة في التفاهم مع الحاسوب بغية معالجة البيانات والحصول على المعلومات المفيدة. وقد كان التركيز في الفصلين التاسع والعاشر على لغة باسكال (PASCAL) كإحدى اللغات التي أثبتت كفاءتها وسهولة استخدامها واعتمادها على مفهوم البرمجة المهيكلة، كما ركز الفصل الحادي عشر على كيفية التعامل مع شاشة باسكال وادخال البرامج وتنقيحها.

لقد وضعنا في نهاية كل فصل عدداً كافياً من المسائل والأسئلة غير المحلولة المتدرجة في صعوبتها رغبةً منا في إتاحة المجال للطالب والقارئ لاختبار مدى فهمه لمحتويات هذا الكتاب.

وعلى الرغم من أن هذا الكتاب عمل جماعي ومسؤولية مشتركة، إلا أن أمانة التأليف تقتضي تخصيص دور كل مؤلف على حدة، فقد قام الدكتور ابراهيم نائب بكتابة الفصول (الأول، الخامس، السادس، السابع، الثامن، الحادي عشر)، وقام الدكتور غازي رحو بكتابة الفصول (التاسع، العاشر)، في حين قام الدكتور محمد ضاهر بكتابة الفصول (الثاني، الثالث، الرابع، السادس).

وإننا إذ نضع هذا الكتاب بين أيدي طلابنا وزملائنا الأعزاء، لا ندعي أننا وصلنا به إلى درجة الكمال، ولكننا نأمل أن تكون المواضيع التي يحتويها هذا الكتاب وطريقة عرضها مفيدة ومفهومة للطالب وجميع المهتمين بالدراسات المعمقة في المواضيع المتقدمة في علم الحاسوب والتي لم يتسع المجال

للخوض في تفاصيلها، وأن نكون قد وفقنا من خلال هذا الجهد المتواضع بوضع لبنة تأسيسيه في بناء مكتبة عربية تزخر بالمؤلفات العلمية في هذا المضمار.

نود هنا أن نتوجه بالشكر إلى جميع من ساهم بشكل مباشر وغير مباشر في إخراج هذا الكتاب، ونرجو من زملائنا الكرام إغناء تجربتنا هذه بملاحظاتهم كي نأخذ بها عند إعادة طباعته، إن شاء الله.

والله ولي التوفيق

المؤلفون

الفصل الأول

الحاسوب (أهميته - نشأته - تطوره)

(1 – 1) أهمية دراسة الحواسيب

إن أهمية دراسة ومعرفة الحواسيب وأنواعها ومكوناها من معدات وأجهزة وبرمجيات ولغات مختلفة يعطي فوائد كثيرة يستطيع انساننا العربي أن يجنيها من تعلم ودراسة هذه المادة التي دخلت في جميع مناحي الحياة ، حيث تساعد هذه المعرفة الإنسان على تكوين مستوى تعليمي رفيع من خلال تعلم الحواسيب وعملها واستخداماها وتساعد في الحصول على إنتاجية عالية وإنجاز دقيق للأعمال وبكفاءة مرتفعة وقدرة على إتمام الكثير من الوظائف والتي قد لا يمكن للإنسان أن ينجزها باستخدام الأيدي أو العقل ، بالإضافة إلى تطوير النظم المختلفة وتجميع قطاعات الحياة ومعالجة البيانات وإدارة هذه البيانات بطرق سهلة وسريعة وتكلفة قليلة ، بالإضافة إلى تحسين النوعية وخاصة في الإنتاج وإدارة العمليات والمعلومات ومعالجة البيانات وكذلك ما يتعلق بالاتصالات التي أصبحت أسلوب مهم وفعال في نقل المعلومات بالإضافة إلى ما يتعلق بالتعليم وتطويره وزيادة معدلات التطور ومدى تأثيره على حياة البشرية وفي جميع القطاعات ، ولهذا فان أهمية دراسة الحواسيب لها تأثير مباشر على حياة الإنسان وتطوره وتقدمه .

(1 – 1 – 1) الحواسيب وأثرها على التربية والتعليم والمعالجة

إن التربية والتعليم هما عاملان أساسيان في تقدم الأمم وإن الشعب المتعلم يخلق حضارة جديدة ويبني الأمم ولهذا فإن الحصول على مستوى تعليمي مرتفع

يمكن أن يأتي من خلال تعلم الحواسيب التي تؤثر على حياة البشر وذلك كونها مستخدمة في جميع قطاعات الحياة إبتداءً من البنوك والمستشفيات والمصانع والطائرات والمؤسسات المدنية والعسكرية وصولاً إلى تحديد الأمراض وتحديد مستقبل الحياة ، لذلك فان تعلم الحواسيب ومعرفتها بصورة جلية والدخول في معرفة العديد من مظاهر التكنولوجيا الموجودة في جميع مناحي الحياة ، ومع تزايد تطور الحياة زادت الحاجة للمعرفة والتعلم بماهية هذا الجهاز وكيفية استخدامه ، ولهذا نرى أن الأمم اتجهت في إدخال الحواسيب في التعليم و التعلم لأبنائها ومنذ المراحل الدراسية الأولى وصولاً إلى شعب متعلم يستطيع معالجة جميع مشاكله من خلال استخدام الحواسيب، إذ أن الحواسيب لا تقوم بتنفيذ بعض المهام اليومية بشكل أفضل وبكفاءة أعلى فحسب ، وإنما يمكن للحواسيب أن تقوم بتنفيذ المهام التي لا يستطيع الإنسان الاعتيادي أن يقوم بها فأنظمة المحاكاة والتنبؤ في الحواسيب تستطيع أن تقدم للإنسان ما لا يستطيع الوصول إليه الإنسان نفسه بالسرعة والجهد المبذول ، فالحواسيب يمكن أن تجري عمليات كيميائية معقده ، والوصول إلى نتائج عالية المستوى دون الخوض في التطبيق من خلال القيام بمزج أنواع مختلفة من العناصر الكيميائية ، وإنما من خلال قيام نظام أو برنامج محدد يحاكي عمليات المزج الفعلية ويعطي نتائج دقيقة وعالية المستوى ، ويستطيع الحاسوب عرض النتائج وتسجيلها وتسلسلها ، ولهذا يستطيع الحاسوب أن يقوم بالتجارب ، ويعطي النتائج الدقيقة ، كما يمكن للحاسوب أن يقوم بمهام عملية التنبؤ للمستقبل ويقوم بعمليات التحليل وإعطاء النتائج وبهذا يستطيع مساعدة الطالب في التعلم والتدريب دون المخاطرة بالمواد والمعدات ، وكذلك فإن تعلم الحواسيب يساعد في تنفيذ وظائف معينة بكفاءة عالية فاستخدام معالج النصوص يعمل على إعداد الوثائق ويساعد في تنظيمها و تسهيلها وسرعتها ودقتها ، وكما وأن المرونة في استخدام الحاسوب تكون دائما أعلى وأكبر من استخدام الآلة الكاتبة مثلاً ،

بالإضافة إلى إمكانية استخدام القاموس الداخلـي Spelling Checker الـذي يساعد كثيراً في تقليل الاخطاء بالإضافة إلى البحث عن الكلمات والذي يساعد في جعل النص المستخدم ذا قوة عالية في المعنى ، هذا بالإضافة إلى أن تعلم اســتخدام الحواسيب تعمل على توفير الوقت والجهد والسرعة .

(1 – 1 – 2) الحواسيب والاتصال مع الغير

Computer and Communication

بعد انتشار استخدام الحواسيب في مختلف المجالات لغرض زيادة وتحسـين الأداء في العمل وزيادة كفاءة الأعمال وسرعتها ، وبتطور ذلك الانتشار وبزيـادة ذلك الاستخدام ازدادت الحاجة إلى تبادل المعلومات والبيانات بـين الوحـدات المكونة لأي مؤسسة أو بين مجموعة مؤسسات ، وحيث كان نقـل البيانـات في السابق وتناقلها والاتصال مع الغير يتم من خلال أسلوب قديم عن طريـق نقـل البطاقات المثقبة أو الأشرطة الممغنطة والتي كان يتم نقلها بالوسائل التقليدية ولكـن مع مرور الوقت تطورت طريقة نقل البيانات وذلك بتطور الاتصالات سواءً كانت هذه الاتصالات شبكات هاتفية أو شبكات بيانية خاصة بالاتصالات الحاسوبية .

وكما هو معلوم فإن المؤسسات الكبرى عندما استخدمت أجهزة الحواسيب على نطاق واسع في هذا القرن لغرض تبادل المعلومات ومعالجة البيانـات تطـور كذلك معها عملية تناقل المعلومات فيما بين هذه المؤسسات حيث كان النمـوذج القائم هو نموذج أجهزة الكمبيوتر المركزية والمحطات الطرفية بحيث يقوم المسـتخدم عادة بإرسال واستقبال المعلومات التي يحتاجها إلى ومن الجهاز المركـزي والـذي يقوم بمعالجة كافة ما يرد إليه من بيانات ، وبناءً على ذلك قامت صناعة كاملة هـي صناعة الأجهزة الايوانية (الكمبيوتر الجبار) والتي كـانت بالنسـبة لزمنـها ذات قدرات معالجة خارقة ، ومع أن هذا النموذج كان يسـمح للمئـات وفي بعـض الأحيان للآلاف من المستخدمين في مشاركة المعلومات و استخدام الكمبيوتـر إلا

أن اكبر معوقات هذا الأسلوب كان ثمن الأجهزة المرتفعة وكذلك بطء الاتصالات و صعوبة الاستخدام ، وبانتشار الحاسوب الشخصي الرخيص الثمن والسريع أخذت المؤسسات تبني نموذجاً جديداً في بناء الاتصالات فيما بينها من خلال بناء الشبكات التي تقوم على مجموعة جديدة من نظم التشغيل التي تتميز نسبياً أيضاً بسهولة الاستخدام ، وفي داخل المؤسسات أيضاً، اعتمدت النماذج المركزية لادارة هذه الشبكات بحيث يتم ربطها بالأجهزة الايوانية أو الحواسيب الصغيرة لغرض الاتصال مع الغير أو الاتصال فيما بين المستخدمين نفسهم لغرض تناقل المعلومات والبيانات المخزنة والاستفادة منها ، وفي خلال التطور الحاصل في نقل البيانات والمعلومات والاتصال مع الغير ، جاء عقد التسعينات حاملاً معه ما عرف أنه أكبر ثورة في نقل المعلومات من خلال الأنترنت والتي استطاعت أن تخلق اتصالاً مع الغير ومع الآخرين والتي ربطت الملايين من أجهزة الحواسيب وساعدت في نقل البيانات والمعلومات وكونت بيئة مؤتمته جديدة تسمح بالمطورين ببناء تطبيقات جديدة تماماً ، من حيث إمكانية الفهم والتعامل مع مختلف الحواسيب دون الحاجة إلى عمليات تحويل من أية نوع كان ، ومع ظهور الأنترنت ظهرت نماذج جديدة للحوسبه مثل جافا (JAFA) وهي لغة برمجة معقدة تمكن من صنع برامج تعمل في أي بيئة تشغيل تدعمها وبهذا تم التوصل إلى ما يعرف الآن بتناقل المعلومات والاتصال مع الغير بسهولة ويسر .

(1 – 2) نبذة تعريفية عن الحاسوب

(1 – 2 – 1) المفاهيم الأساسية للحاسوب

شهدت السنوات الأخيرة انتشاراً واسعاً لاستخدامات الحاسوب في جميع مناحي الحياة، فقد فرض الحاسوب نفسه بما يمتلك من الإمكانات الهائلة على سائر العلوم ، وذلك بسبب تزايد حجوم البيانات والمعلومات المرافقة للتطور العلمي والتقني المستمر في مختلف الميادين ، هذا التزايد ألزم الإنسان اللجوء إلى نظم المعالجة

الآلية الحديثة ، بدلاً من الأنظمة اليدوية، والمتمثلة بالأنظمة الحاسوبية التي أصبحت في وقتنا الحالي أداة فعالة وأساسية في حفظ ومعالجة البيانات مهما يكن حجمها وبالسرعة القصوى .

ولكي نستطيع تسخير إمكانيات الحاسوب في أي مجال من مجالات الحياة بهدف مواكبة التطور العلمي والتكنولوجي لا بد من توفر عناصر أخرى مساندة ومدعمة لعمل الحاسوب كجهاز ، وهذا ما ستدعي إلى إنشاء نظام حاسوبي الذي يتألف من العناصر الأساسية التالية :

أولا : الأجهزة Hardware

وهي عبارة عن المكونات المادية والتي تشتمل على جهاز الحاسوب والأجهزة المساعدة له . إن التطور السريع في عالم الحواسيب أدى إلى افراز تشكيلة واسعة ومتنوعة من الحواسيب تتراوح من حواسيب مايكروية زهيدة الثمن إلى أخرى كبيرة ومعقدة غالية الثمن ، إلا أن جميع الحواسيب على اختلاف حجمها وتصنيفها والمهام التي يكمن أن تؤديها ، تشترك في عدد من الخصائص والمواصفات، إذ أن جميعها تتكون من الوحدات التالية انظر الشكل (1) :

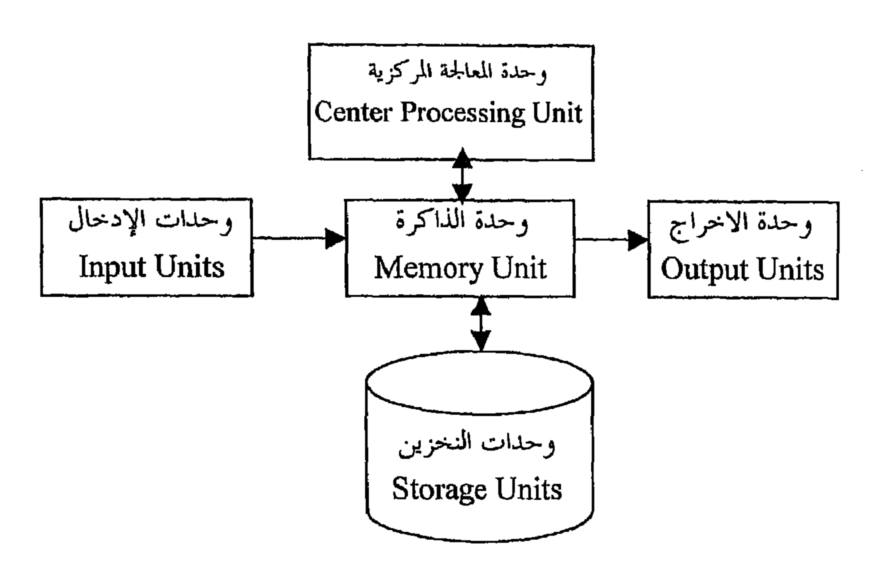
1 – وحدة المعالجة — Central Processing Unit

2 – الذاكرة — Memory

3 – وحدات الإدخال و الإخراج — Input / Output Units

4 – وحدات التخزين المساعدة — Storage Units

و سنأتي على شرح المكونات المادية إن شاء الله في فصول لاحقة .



الشكل (1) الهيكل العام لمكونات الحاسوب

ثانياً : البرمجيات Software

هي مجموعة الأوامر أو التعليمات المتسلسلة التنفيذ و التي تستخدم في إدارة ومراقبة وتشغيل جهاز الحاسوب ، هذه الأوامر والتعليمـــات تســمى البرنــامج (Program) وتصنف عادة البرمجيات إلى نوعين :

1. **برمجيات النظم System Software :**

وهي مجموعة البرامج تساعد في تشغيل نظام الحاسوب وتعمل على توجيـــه أجهزة الحاسوب المختلفة لاداء الوظائف الأساسية بها عند تنفيـــذ برنــامج المستخدم بشكل تلقائي.

2. **البرمجيات التطبيقية Application Software :**

وهي البرامج المعدة لخدمة المستخدم في إحدى المجالات مثل برنامج (الــوورد) (Word) (اكسل Excel) (برامج المحاسبة) (برامج تحليل المبيعات ومراقبــة المخزون) (برامج الرسم الهندسي) ...

وسنتعرف على المزيد من البرمجيات عندما تتناول فصل البرمجيات بشكل مفصل إن شاء الله .

ثالثاً : البيانات Data

وهي مفاهيم لغوية أو رياضية أو رمزية متفق عليها لتمثل الأحداث أو الأشخاص أو الأشياء ، هذه المفاهيم تحتاج إلى معالجة Processing كي تتحول إلى معلومات مفيدة لمن يحتاجها ، والحاسوب لديه القدرة على هذه المعالجة بسرعة كبيرة ودقة متناهية مهما يكن حجم البيانات المراد معالجتها ، وهذا ما أعطى للحاسوب هذه المكانة المرموقة بين سائر العلوم .

رابعاً : العنصر البشري Users

وهم مستخدمي نظام الحاسوب والمستفيدون من تشغيله ويتوقف نجاح نظام الحاسوب وقدرته إلى حد كبير على كفاءة العنصر البشري وتمكنه من استخدام الحاسوب وقدرته على تسخير هذا الجهاز بواسطة البرمجيات المتاحة والبيانات المتوفرة في تحقيق أهدافه، وبسبب توسع وتطور علم الحاسوب أصبح هناك متخصصين في شتى أقسامه وبالتالي يكن تصنيف العنصر البشري إلى فئات مثل :

مبرمج Programmer ، محلل نظم Processing Analysis

مهندس برمجيات Software Engineer ،

الذي يقوم بإدخال البيانات Typist ، مشغل الحاسوب Operator ،

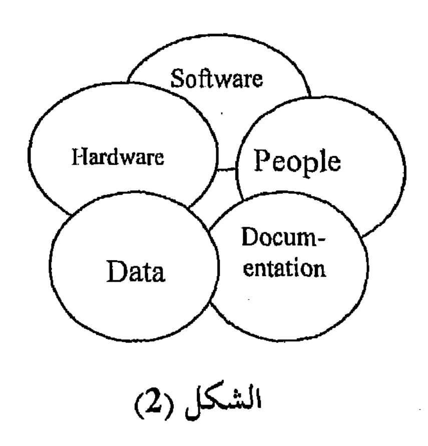
الفني الذي يقوم بفحص أجزاء الحاسوب وتغيير الأجزاء المعطلة .

خامساً : التعليمات والوثائق Documentation

و هي مجموعة التعليمات الموثقة والتي توضح للمستخدم كيفية استخدام النظام الحاسوبي وتشغيله لتنفيذ التطبيقات المختلفة المطلوبة،

والإجراءات اللازمة اتباعها عند حدوث أي عطل أو خلل في أداء النظام الحاسوبي .

إن العناصر السابقة ترتبط بشكل وثيق مع بعضها البعض وأي قصور في أي عنصر منها يؤدي إلى قصور عمل النظام الحاسوبي ، ويمكن تمثيل الترابط بين العناصر كما في الشكل (2) .



الشكل (2)

(1 – 2 – 2) تعريف الحاسوب وخصائصه

إن كلمة كمبيوتر (Computer) مشتقة من فعل يحسب Compute ، لذلك عربت كلمة كمبيوتر في اللغات الأجنبية إلى كلمة حاسوب في اللغة العربية ويعرف الحاسوب بتعاريف مختلفة نذكر منها :

- الحاسوب : هو آله حاسبة إلكترونية ذات سرعة عالية جداً ، ودقة متناهية يمكنها قبول البيانات وتخزينها ومعالجتها للوصول إلى النتائج المطلوبة .
- هو نظام إلكتروني لمعالجة البيانات وفقاً لمجموعة من التعليمات .
- هو مجموعة من الأجهزة الإلكترونية تدعىHardware يتم التحكم في أدائها بواسطة مجموعة من البرامج المخزنة تدعى Software .

ومن هذه التعريفات واستناداً إلى عمل النظام الحاسوبي يتضح لنا أن الحاسوب ليس مجرد جهاز أو آلة ولكنه نظام متكامل يتكون من مجموعة الأجهزة

Hardware و مجموعة التعليمات المخزنة Software ، مدخلات هذا النظـــام هي البيانات Data التي يقوم الحاسوب بمعالجتها أو تخزينـــها أو كليـــهما ، أمـــا مخرجاته فهي المعلومات Information أو النتائج Results . ولا يتم ذلك كلـه بمعزل عن الإنسان المشغل والمستخدم والمستثمر لهذا النظام، أي يتم تحت إشـــرافه وإدارته ومراقبته وتخطيطه الكامل .

ويتميز الحاسوب بإمكانيات وقدرات خاصة نذكر منها :

1. **السرعة الفائقة في أداء و تنفيذ التعليمات :**

يقوم الحاسوب بالإضافة إلى التعليمات الحسابية المعروفة ، بمقارنة الحـــروف حسب تسلسلها الهجائي وكذلك تحرير ونقل ونسخ وتخزين الأرقام والحـــــروف والرموز والكلمات بسرعة فائقة ، إذ أن الحاسوب الصغير يمكنـــه تنفيـــذ مئـــات الآلاف من التعليمات في الثانية ، بينما الحاسوب الأكبر يمكنه تنفيذ عدة ملايـــين، لذلك اصبح بالإمكان في هذه الأيام من معالجة كمية هائلة من البيانات والحصـــول على كم من المعلومات ، كان من العسير الحصول عليها منذ سنوات قليلة مضـــت إلا بعد مجهود بشري كبير يستغرق وقتاً طويلاً.

2. **الدقة في تنفيذ العمليات المختلفة :**

بالإضافة إلى السرعة الفائقة فان الحاسوب يعطي النتائج بدقة متناهيـــــة، إذا كانت البيانات المدخلة صحيحة وإذا كان البرنامج الذي يقوم بمعالجـــة البيانـــات صحيحاً ومنطقياً ، وكلاهما البيانات المدخلة والبرنامج من صنع الإنسان فإذا كـــان هناك قصوراً في أحدهما فان النتائج ستكون غير صحيحة .

3. **القدرة على العمل فترات طويلة دون أخطاء :**

يتكون الحاسوب من دوائر إلكترونية، وهي ليست عرضة للتـــــآكل مثـــل المكونات الميكانيكية ، ونظرا للتطور الهائل في تكنولوجيا تصنيع الإلكترونيات، فإن الدوائر الإلكترونية أصبحت ذات كفاءة عالية جداً ، إذ بإمكانها أداء مئات الآلاف

أو الملايين من العمليات كل ثانية ولفترات طويلة دون تعب أو ملل فإذا كان هناك 5 ملايين عملية حسابية يجب إجراءها فان الحاسوب ينفذ العملية رقـــم الـــ 5 مليون بنفس الدقة والسرعة تماماً كما ينفذ العملية الأولى .

4. تعدد الاستعمال :

حيث يستطيع الحاسوب حل كثير من المشاكل التي تواجه الإنســـان مـــن خلال أنواع متعددة من البرامج مثل برامج معالجة النصوص ، برامـــج النوافـــذ ، برامج قواعد البيانات...

5. الكفاءة العالية في إدارة البيانات :

بمجرد إدخال البيانات وتشغيلها يقوم الحاسوب بتنفيذ أحد أو بعـــض أو كل العمليات التالية :

- التخزين لحفظ البيانات والمعلومات لحين الحاجة إليها .
- الاسترجاع أي استعادة البيانات والمعلومات المخزنة لإعـــادة اســتخدامها أو الاطلاع عليها عند الحاجة .
- نقل المعلومات من موقع إلى آخر عبر قنوات اتصال لاستخدامها أو لإجـــراء المزيد من عمليات التشغيل والمعالجة لها حتى تصبـــح في الصـــورة المطلوبـــة للمستخدم النهائي .
- نسخ المعلومات أو إعادة إنتاجها ، لإنشاء نسخة احتياطيـــة مـــن البيانـــات والمعلومات .

6. تنفيذ العمليات المطلوبة آلياً :

يقوم الحاسوب عادة عن طريق البرنامج المصمم له بأداء عمليات متعـــددة دون تدخل الإنسان ، إذ أن تركيب البرنامج وهيكليته تساعد الحاســـوب علـــى الانتقال بين التعليمات بسرعة كبيرة وبصورة آلية .

رغم الخصائص والمميزات السابقة للحاسوب ، هناك عددا من الأمور تحـــد

من إمكانيات وقدرات الحاسوب منها :

1- انعدام الذكاء الفطري :

وذلك أن الحاسوب لا يفكر، وينفذ ما يطلب منه من أوامر وتعليمات غير مذكورة في البرنامج.

2- تتوقف كفاءته على كفاءة ودقة البرامج :

لأنه كما ذكرنا لا يمكن للحاسوب التفكير وبالتالي كلما كانت البرامج التي تضم التعليمات والأوامر دقيقة ومنطقية، كلما كان عمل الحاسوب أفضل، أضف إلى ذلك فان البرنامج الجيد لا يعطي النتائج المطلوبة إلا إذا تم تغذية الحاسوب بالبيانات اللازمة والصحيحة والموثوقة .

3- النظام الحاسوبي أقل مرونة من النظام اليدوي وذلك لأن أي خلل في أي عنصر من عناصره يتطلب وقتا وجهدا لاصلاحه .

(1-2-3) أنواع الحواسيب

إن ازدياد الحاجة إلى استخدام الحواسيب في شتى المجالات وتنوع استخدامها، أدى إلى ظهور أنواع مختلفة منها ، إذ ظهرت الحواسيب المترلية والحواسيب المكتبية والحواسيب للاستخدامات الإدارية والحواسيب لاستخدامات مراكز البحوث ومراقبة الإنتاج والتحكم به . وغيرها من الأجهزة ، أضف إلى ذلك اختلفت تلك الأنواع بحجمها واتساع ذاكرتها وطريقة معالجة البيانات. لذلك يمكن تصنيف الحواسيب إلى ثلاثة أنواع من حيث العمل :

النوع الأول : حسب أسلوب معالجة البيانات :

وتصنف إلى ثلاثة أنواع أيضاً :

1- حواسيب رقمية Digital Computers :

هذا النوع من الحواسيب يعالج البيانات بتحويلها إلى أرقام حسب نظام

عددي معين، ثم يقوم بمعالجتها حسابياً ومنطقياً ومن ثم تحويل الأرقام المعالجـــة إلى المعلومات المطلوبة، وتستخدم الحواسيب الرقمية في المجالات العلميـــة والتجاريـــة والاجتماعية.

2- حواسيب قياسية Analog Computers :

هذا النوع يعالج البيانات باستخدام طرق القياس التي تعتمد على الظواهـــر الفيزيائية مثل الضغط الجوي ودرجة الحرارة و الجهد الكهربائي،.. وهذا النوع من الحواسيب تستخدم في مجال الدراسات العلمية و مجالات التحكم الآلي.

3- حواسيب مهجنه Hybrid Computers :

هذه الحواسيب تجمع بين خصائص الحواسيب الرقمية و الحواسيب القياسـية وتستخدم في بعض المجالات العلمية الخاصة مثل أبحاث الفضاء و الاستشعار عـــن بعد.

النوع الثاني : حسب الغرض من استخدامها :

و تصنف إلى نوعين :

1- حواسيب ذات الغرض الخاص Special- Purpose Computers

و هي حواسيب مصممة لتنفيذ غرض معين والقيام بوظيفة محـــددة مثـــل الحواسيب الخاصة بالطيران الآلي – الحواسيب المســـتخدمة لأغـــراض عســـكرية كتوجيه الصواريخ والتحكم بشبكات الرادار – الحواسيب المستخدمة في الصناعـــة للتحكم بخطوط الإنتاج ....

2- حواسيب ذات الغرض العام General–Purpose Computers

وهي حواسيب مصممة لتنفيذ العديد من المهام في المجالات العلمية المختلفـــة والأعمال الإدارية ، وذلك باستخدام برمجيات مختلفة حسب نوع المشكلة المـــراد معالجتها .

النوع الثالث : حسب حجم الذاكرة :

وتصنف إلى خمسة مجموعات :

1. الحواسيب العملاقة : Super Computers

تتميز بوجود ذاكرة ضخمة ، سرعتها كبيرة جداً و لها اكثر من معالج واحد . استخدامها محدود جداً بسبب كلفتها الباهظة .

في عام 1981 أعلنت مجموعة من الشركات المنتجة للحواسيب العملاقة عن إنتاج حاسوب أسموه (الحاسوب الميكرو العملاق Supermicro Computer ، حيث قدراته تقترب من الحاسوب العملاق و ذاكرته تزيد عن 128 ميجابايت .

2. الحواسيب الكبيرة Mainframe Computers

تتميز بوجود ذاكرة رئيسية كبيرة يمكن توسيعها ، وتستخدم مع هذه الحواسيب أجهزة طرفية تصل إلى مئات من الأجهزة ، بالإضافة إلى أن سرعة وحدة المعالجة المركزية كبيرة ، لذلك تستخدم بشكل واسع في الشركات الكبيرة مثل شركات الطيران و إدارات الهجرة و الجوازات وهيئات الدفاع...

إلا أن مساوئها كثيرة منها أنها تحتاج إلى مصدر خاص للطاقة و نظام للتهوية و تحتاج إلى عدد من الموظفين لتشغيلها و إدارتها ذوي خبرة واختصاص في الحواسيب الكبيرة بالإضافة إلى ارتفاع ثمنها .

3. حواسيب الميني Mini Computers

أهم ميزاتها وجودها في صورة عائلة بحيث يمكن للمستثمر استبدال حواسيبه بحواسيب أعلى من نفس العائلة و بالسعر المناسب كلما دعت الحاجة، بالإضافة إلى أنها تتيح تعدد المستخدمين يصل إلى مائة مستخدم أو اكثر ، لذلك تستخدم بشكل واسع في المؤسسات والمنظمات التي تتطلب قدرات كبيرة في تخزين البيانات و معالجتها إلا أن سعرها مرتفع .

4. حواسيب الميكرو Micro Computers

و هي أصغر الحواسيب حجماً وأقلها تكلفة و يوجد في الأسواق الآن العديد من الأنواع المختلفة حسب حجم ذاكرتها و نوع معالجها وسرعة أداء العمليات ، نذكر منها :

- الحواسيب المنزلية Home Compute
- الحواسيب المحمولة Portable Computers
- الحواسيب المكتبية Desktop Computers
- الحواسيب الشخصية Personal Computers

و تعتبر الحواسيب الشخصية أكثر الحواسيب انتشاراً ، تستخدم في مجالات الأعمال الصغيرة والمتوسطة بالإضافة إلى استخداماتها الشخصية المختلفة، وذلك لسهولة استخدامها و توفر البرمجيات التطبيقية لها .

نشير هنا إلى أن هذه الحواسيب في تطوير مستمر من حيث حجم ذاكرتها و سرعة معالجها وأسلوب التعامل معها .

(1 – 2 – 4) استخدامات الحواسيب

في الوقت الراهن يقاس مدى تطور وتقدم أي مؤسسة أو شركة بمدى استخدامها للحاسوب و بحجم وسرعة الخدمات التي تنجزها أجهزتها الحاسوبية ، إذ أن الحاسوب دخل جميع ميادين العلم و المعرفة بسبب الحاجة الماسة لدى جميع الأوساط إلى الحصول على نتائج متناهية في الدقة و سرعة في الإنجاز و المعالجة للبيانات الكثيرة و المتزايدة يوماً بعد يوم ، حتى أنه أصبح من الصعب أن نجد اليوم علماً من العلوم لم يتدخل الحاسوب فيه، ومن الصعب أن نحصر جميع الاستخدامات البشرية لهذا الحاسوب المتميز بالإمكانيات والقدرات ، إلا أننا سنوجز بعض الاستخدامات الشائعة له :

- الشؤون الإدارية و تنظيم ملفات الموظفين و المعدات وحفظ جميع البيانات الإدارية المتعلقة بالدوائر .
- الدراسات المالية و المحاسبية و الرواتب .
- المبيعات و التسويق و الترويج على البضائع و لاسيما حاليا من خلال شبكة الأنترنت .
- الدراسات الإحصائية و ذلك لمعالجة البيانات الإحصائية أثناء عمليات التعداد السكاني وعمليات المسح الجغرافي و الديموغرافي بالإضافة إلى تحليل البيانات واستخدامها في مجالات التنبؤ و تهيئة المعلومات اللازمة للدراسات الاقتصادية بمختلف أنواعها .
- أعمال السكرتارية و الطباعة ، و تهيئة المستندات و العقود اللازمة بشكل فريد من نوعه، واعداد الكتب و المراجع العلمية و اخراجها بشكل لائق .
- في تنظيم أعمال المخازن والمستودعات و تهيئة الطلبيات اللازمة للاستيراد والتصدير .
- اتمتة المكاتب والبريد الالكتروني .
- أعمال البنوك و البورصات المالية .
- في مجالات تخطيط الإنتاج و الرقابة عليه باستخدام طرق علمية وأساليب مدروسة .
- في مجالات تنظيم السفر وحجز أماكن المسافرين على الطائرات والقطارات والباصات ... وتهيئة البيانات حول موعد وصول أو مغادرة وسائل النقل .
- في مجالات العلوم الطبية فقد صار بإمكان الحاسوب التحكم بغرف العمليات وأجهزة التصوير المختلفة و الأجهزة الطبية المنظورة وتهيئة جميع البيانات المتعلقة بالمشكلة المعالجة، بالإضافة إلى ربط المستشفيات المعالجة مع بعضها لخدمة الطب و الاستشارات الطبية وتبادل الخبرات الطبية .

- في مجالات الهندسة فقد اصبح الحاسوب الوسيلة الأساسية في إجراء الحسابات اللازمة لإنشاء وبناء المشاريع الهندسية ، وفي وضع التصاميم الهندسية والتحكم بها إذ بالإمكان إنشاء العديد من التصاميم لاختيار أفضلها ...
- في العلوم العسكرية و خاصة في دراسة الإستراتيجيات العسكرية للجيوش وتحديد مواقع و إمكانيات العدو من خلال تحليل كم من البيانات وإعطاء الحلول المثلى بسرعة و دقة فائقة معتمدا على العلوم العسكرية – الرياضية و من أهمها نظرية الألعاب الإستراتيجية Games theory .
- في مجالات التعليم فقد ساهم و يساهم في تطوير التعليم من خلال برامج تعليميه موجهة للاطفال و الكبار ، و من خلال مساعدة الطلبة و المدرسين على التزود بالمعرفة و من خلال الخدمات العلمية التي يقدمها عبر شبكات الأنترنت .بالإضافة إلى كونه اصبح أداة لتطوير و تحديث الكثير من الأساليب و الطرق الرياضية و الفيزيائية و الكيميائية .
- في مجال الفلك فقد أصبح الحاسوب يتحكم في توجيه الأقمار الصناعية ورصد دوران الكواكب و المجرات و اكتشاف المزيد من خفايا الكون ...

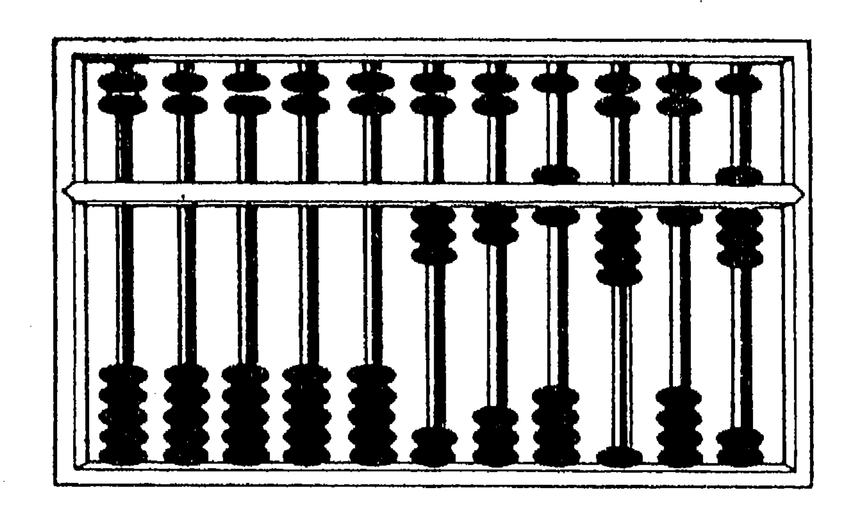
(1– 2 – 5) لمحة تاريخية عن تطور الحواسيب

إن اختراع الحاسوب لم يأتي من فراغ ، بل كان نتيجة لمحاولات و أفكار عديدة سابقة بهدف تسهيل إجراء الحسابات وتطوير وسائل معالجة البيانات التي صار يزداد حجمها يوماً بعد يوم، لذلك سنعرض في هذه الفقرة الأفكار والآلات التي أدت وكانت السبب في وصول الحاسوب إلى ما هو عليه في الوقت الحاضر .

1– استخدام الحصى لمعرفة كمية الأشياء ثم استعمال أصابع اليد في العد .

2– المعداد Abacus :هو أول وسائل الحساب اليدوي أستخدم لإجراء العمليات الحسابية الأربع ، و يتألف من إطار خشبي يحوي عدداً من الأعمدة، كل عامود يضم عدد من الكرات أو الخرزات مستديرة الشكل على الأعمدة.

الشكل (3) ، يعود اكتشاف العداد إلى عام 2000 ق.م في كل من الصين واليابان و روسيا ، واستخدم خلال الحضارات القديمة ، و استمر مسيطراً مئات من السنين ، فقد شاع استخدامه في أوروبا في القرن السادس عشر الميلادي ،ومازال مستخدما حتى الآن في بعض البلدان مثل روسيا و اليابان .



الشكل (3) أحد أشكال المعداد

3- أعمدة نابير Napiar's Roods and Bones

وضع هذه الوسيلة العالم الرياضي جون نابير (John Napiar) عام (1617) و هي عبارة عن جداول رياضية تعمل على أساس اللوغاريتمات لتحويل عملية الضرب إلى عملية جمع و فق أسلوب معين ، صنعت على هيئة شرائح خشبية كل منها مقسمة إلى تسعة مربعات صغيرة وكل مربع يقسم إلى قسمين ، كل عامود يمثل جدول الضرب للأرقام من 1وحتى 9 بحيث يكتب رقم الآحاد اسفل قطر المربع أما رقم العشرات فيكتب أعلى القطر الرئيسي الشكل(4).

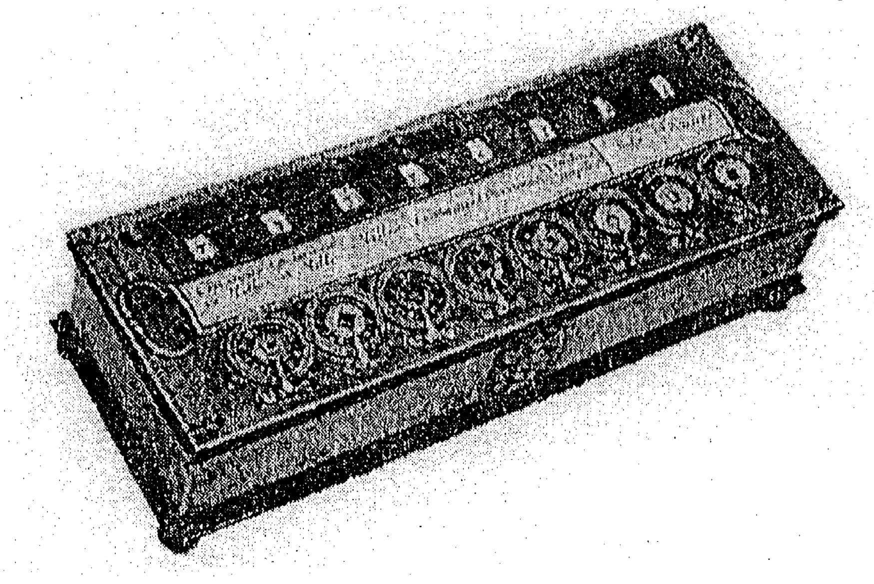
1	2	3	4	5	6	7	8	9	0
2	4	6	8	1/0	1/2	1/4	1/6	1/8	0
3	6	9	1/2	1/5	1/8	2/1	2/4	2/7	0
4	8	1/2	1/6	2/0	2/4	2/8	3/2	3/6	0
5	1/0	1/5	2/0	2/5	3/0	3/5	4/0	4/5	0
6	1/2	1/8	2/4	3/0	3/6	4/2	4/8	5/4	0
7	1/4	2/1	2/8	3/5	4/2	4/9	5/6	6/3	0
8	1/6	2/4	3/2	4/0	4/8	5/6	6/4	7/2	0
9	1/8	2/7	3/6	4/5	5/4	6/3	7/2	8/1	0

الشكل (4) أعمدة نابير

4- آلة باسكال Pascal Machine

و هي آلة ميكانيكية للجمع و الطرح اخترعها العالم الرياضي الفرنسي بليز باسكال (Plaise Pascal) عام 1642 بهدف تسهيل عملية مسك الدفاتر المحاسبية في جمع الضرائب لوالده .والآلة عبارة عن مجموعة عجلات كل عجلة مقسم إلى عشرة مقاطع مسجل عليها الأرقام من 1 إلى 9 وتتصل العجلات مع بعضها البعض بحيث تدور العجلة الثانية حركة واحدة عندما تدور العجلة الأولى دورة كاملة و هكذا بالنسبة لباقي العجلات، حيث أن العجلة الأولى تمثل منزلة الآحاد والعجلة الثانية تمثل منزلة العشرات وهكذا، الشكل (5)، تعتبر هذه الآلة الأساس في تصميم الآلات الميكانيكية التي ظهرت فيما بعد و حتى منتصف

الستينات، وتكريما لاسم العالم باسكال أطلق اسمه على إحـــدى لغـــات برمجـــة الحاسوب (لغة باسكال) .



الشكل (5) آلة باسكال

5- آلة ليبيتز (Leibiniz Machine)

تعتبر هذه الآله تطوير لآلة باسكال تقوم بعمليـــات الضـــرب و القســـمة واستخراج الجذور بالإضافة إلى عمليات الجمع والطرح، صممها العـــالم الالمـــاني جوترفريد وليام ليبنيز Gotrfried wilhelm Leibniz عام 1671 .

6- آلة جون ميلر J. H. Muller حيث اقترح العالم جون ميلر 1786 بنـــاء حاسبة تقوم بحساب الدوال الرياضية بشكل تلقائي.

7- آلة الفروقات Difference Machine

بنى هذه الآلة العالم الإنجليزي شارل بابلـج Charles Babbage عـــام 1922 اعتماداً على فكرة العالم ميلر لحساب الدوال الرياضية بشكل تلقائي و بنـاء الجداول اللوغريتمات باستخدام طريقة الفروقات و سميت بآلة الفروقـــات إلا أن هذه الآلة بنيت بشكل جزئي بسبب التكاليف المالية الباهظة .

8- الآلة التحليلية Analytical Machine

صممها العالم باباج عام 1937 على نفقته الخاصة ، وهي آلـــه متطــورة تحتوي الكثير من مواصفات الحاسوب الرقمي العصري مثـــل وحــدة الذاكــرة ووحدات الإدخال والإخراج .

9- نظام البطاقات المثقبة

يعتبر العالم هرمان هوليرث Herman Hollerith (1860–1929) أول من وضع نظام البطاقات المثقبة لتخزين البيانات ضمن قواعد محددة، حيث صمــم آلة لترتيب البطاقات و آلة لوضع البيانات في جداول .

10- الآلات الكهروميكانيكية Electromechanical Machine

لم تتوقف الأبحاث و النظريات حول تصميم أجهزة تخزين ومعالجة البيانــات سوى أثناء الحرب العالمية الثانية ، بعدها قام العـــالم الأمريكــي هــوارد ايكــن Howard Aicon الأستاذ في جامعة هارفارد بالولايـــات المتحـــدة الأمريكيــة بتصميم أول حاسوب كهر وميكانيكي معتمداً على أفكـــار وتصـــاميم العـــالم الإنجليزي باباج وقامت شركة IBM الأمريكية بتنفيذ هذا التصميم عـلم 1944 و سمي بمارك –1 (Mark-1) أهم ميزاته :

1. طوله 51 قدما .
2. عدد مفاتيحه 300 مفتاح .
3. ارتفاعه 8 أقدام .
4. طول أسلاكه 500 ميل .
5. عدد قطعه اكثر من مليون .
6. بطئ في سرعة العمليات الحسابية .
7. طريقة تشغيله معقدة جدا إذ بلغ عدد صفحات دفـــتر التعليمــات الخاصة بتشغيله (547) صفحة يجب إتقانها من قبل كل مستخدم.
8. يستقبل البيانات على بطاقات مثقبة .
9. يقوم بالعمليات الأربع و حساب اللوغريتمات و النسب المثلثية .

11- الآلات الإلكترونية Electronic Machines

- في عام 1946 قام العالمان ايكارت J. P. Echert و موشلي J. W. Mauchly الأستاذان في كلية مور للهندسة في جامعة بنسلفانيا بالولايات المتحدة الأمريكية بتصميم أول حاسوب إلكتروني رقمي باسم: ENIAC (Electronic Numerical Integrator and Automatic Calculators) وقامت شركة بل للتليفونات (Bell Telephone Co.) بتنفيذه ، في صناعته 18 ألف أنبوب مفرغ تعمل على أساس النبضات الكهربائية ، و قد استخدم هذا الجهاز في التعداد العام للسكان في الولايات المتحدة الأمريكية إذ كان لهذا الاستخدام الأثر الكبير في مجال تهيئة البيانات الإحصائية .

-خلال الأعوام (1946 – 1949)

اقترح العالم الأمريكي (J. V. Neuman) بناء حاسوب يخزن البيانات والتعليمات سرية على شكل شيفرات معينة ، ودعيت فكرته بمبدأ تخزين البرامج وطبقت فكرته على الحاسوب EDVAC : Electronic Discrete Variable Automatic Computer ، ثم قام العالم ويلكس (M.V.Wilkes) من جامعة كمبريج – انجلترا بتطوير الحاسوب السابق و أسموه (EDSAC) (Electronic Delay Storage Automatic Computer) حيث استخدم فيه ذاكرة لحفظ البرامج .

-عام 1952 ظهر حاسوب باسم UNIVAC (Universal Automatic Computer) لخدمة معالجة البيانات الإدارية مستخدماً أنابيب كهربائية مفرغة ، و هو أول حاسوب إلكتروني ظهر على المستوى التجاري بهدف التسويق من قبل شركة راند .

-بعد ذلك انتشرت صناعة الحواسيب الإلكترونية بشكل واسع فظـــهرت حاســبات شـــركة (IBM) وشـــركة (NCR) و شـــركة (ICL) الانكليزية و غيرها من الشركات .

وحتى يومنا هذا ما زال السباق على قدم وســاق في عـــالم تطويـــر الحواســيب واستخداماتها .

(1 – 2– 6) أجيال الحواسيب Computer Generations

نتيجة للتطور المستمر لأجهزة الحواسيب منذ عام 1944 (منذ اختراع أول حاسوب (Mark – 1)) وحتى يومنا هذا ، فقد صنـــف المختصـــون في عـــالم الحوسبه أجهزة الحواسيب ضمن فترات زمنية دعوها الأجيال Generations .

أولا : حواسيب الجيل الأول First Generation Computers

وتشمل على الحواسيب التي أنتجت خلال الفترة الزمنيـــة 1945 - 1959 وتدعى حواسيب الأنابيب المفرغة . وقد تأثرت حواسيب هذا الجيـــل بالأعمـــال البحثية الرائدة والتي نذكر منها ثلاثة أعمال هامة :

العمل الأول : للباحث آلآن تورينج Alan Turning الذي صمم أول نمـــوذج للآلات المنطقية عام 1926و صمم أيضا أول اختبار لذكـــاء الآلات الذي يطلق عليه اختبار تورينيج Turning test.

العمل الثاني : للباحثـــان الفرنســيان كوفيجنـــال (Couffignal) و فالتـــان (Valtan) اللـــذان نشـــرا في عـــام 1936 بحثـــا عـــن بعـــض الأفكار النظرية لتطبيق النظام الثنائي في بنـــاء الحاســـوب.

العمل الثالث : فيعود للعالم الرياضي جون فون نيـــومـــان (J.V.Neuman) وآخرون في توثيق المبادئ الأساسية للحاسبات الرقمية عـــام 1945 من خلال مسودة تقرير عن الحاسوب EDVAC ، حيث وضحـــوا في التقرير إمكانية استخدام النظام الثنائي في بناء الحاسبات و طريقـــة

استخدام الدوائر الرقمية على التـوازي في الوحـدات الحسـابية و اختبارها ، وإمكانية تخزين البرامج و البيانات في ذاكرة الحاسوب.

ومنذ عام 1938 وحتى عام 1959 أنتجت العديد من الحواسيب المختلفـة في ألمانيا وأمريكا وبريطانيا .. تلبي حاجة مخترعيهم لأغراض مختلفة منها الحواسيب ENIAC , EDVAC , EDSAC , UNIVAS .

و من أبرز الخصائص و الملامح التي تتميز بها حاسبات الجيل الأول هي :

1. الاعتماد على تكنولوجيا الصمامات المفرغة Vacuum Tubes، مما أدى إلى ضخامة المعدات و كبر حجمها ، و استهلاكها طاقة كهربائية كبيرة .
2. الحرارة الناتجة عن تشغيل الحاسبات تزداد مع طيلة فترة الاستخدام مما يتطلب تهيئة المناخ المناسب من حيث التبريد و الرطوبة لضمان اسـتمرارية العمـل ، وإلا يجب إيقاف الأجهزة عن العمل بين الحين و الآخر .
3. سرعة إنجاز العمليات بطيئة (حيث لا يتجاوز 1000 عملية في الثانية) ، فيما إذا قورنت بحاسبات الأجيال اللاحقة بالإضافة إلى سعة ذاكرتها المتواضعة.
4. غالية الثمن مما اضطر الناس إلى استئجارها بدلا من شرائها .
5. استخدم في برمجة حواسيب هذا الجيل لغة الآلة Machine Language ، ثم استخدم لغة التجميع Assembly Language في نهاية هذا الجيل .
6. استخدمت حواسيب هذا الجيل في عملها أسلوب الذاكرة المعنونة ، حيـث اعتمد عمل الذاكرة في بادئ الأمر على أشعة الكاثود (Cathode Ray) ثم استبدلت بالذاكرة ذات الخلايا الممغنطة
7. أجهزة الجيل الأول كانت تعمل على أساس نظام البرنـامج الواحـد (أي تنفيذ برنامج واحد في كل مرة) .
8. استخدمت البطاقات المثقبة و الأشرطة الورقية المثقبة في تخزين البيانات .

ثانياً : حواسيب الجيل الثاني :Second Generation Computers

وتشمل الحواسيب التي أنتجت من منتصف الخمسينات و حتى منتصف الستينات أي ضمن الفترة الزمنية 1959 و حتى 1964 وسميت بحواسيب الترانزستورات و قد تميزت حواسيب هذا الجيل بتطوراً كبيراً في مجال المكونات المادية Hardware و البرمجيات Software ففي مجال المكونات المادية :

1 - استخدام الترانزستور بدلاً من الأنابيب المفرغة (حيث مساحة الأنبوب المفرغ يمكن أن يحتوي على 200 ترانزستور) مما أدى إلى صغر حجم الحواسيب المنتجة و تقليل من استهلاك الطاقة الكهربائية اللازمة للتشغيل ، انعكس ذلك على التخفيف إلى حد ما من انتشار الحرارة الناتجة عن التشغيل.

2 - زيادة سرعة الأداء حيث أصبحت تقاس بالميللي ثانية (1/1000 من الثانية).

أما في مجال البرمجيات :

1. بدأ استخدام اللغات ذات المستوى العالي (High Level Language) ومنها لغة الفورتران (Fortran) بالإضافة إلى استخدام لغة التجميع في برمجة الحواسيب .

2. بدأ استخدام الأقراص الممغنطة كوسيط لتخزين البيانات والمعلومات باستخدام أسلوب الوصول العشوائي ، هذا ما تم في الحاسوب (RMAC 305) حيث استخدم فيه 50 قرصاً ممغنطاً لتسجيل 5 مليون حرف مع إمكانية الوصول إلى أي سجل في زمن اقل من ثانية واحدة .

ومن الجدير بالذكر انه في نهاية هذا الجيل بدأ عدد من الشركات الصغيرة في إنتاج حواسيب قليلة التكلفة أطلق عليها حواسيب الميني

Low-Cost Minimal Computers (Mini)

ثالثاً : حواسيب الجيل الثالث : Third Generation Computers

أطلق عليها أسم حواسيب الدارات المتكاملة وتدخل ضمن الفـــترة الزمنيـــة 1964 وحتى عام 1971 وتعتبر عائلة الحواسيب (IBM / 300) من شـــركة IBMأول حواسيب هذا الجيل والتي استخدمت في المجالات العلمية والتجاريـــة في آن واحد، من أهم الملامح والخصائص التي تميزت به حواسيب هذا الجيل :

1- استخدمت تكنولوجيا الدوائر المتكاملة (Integrated circuits) ممـا أدى إلى صغر حجم الحواسيب ووزنها ، وأصبحت تكلفتها أقل وكفاءتها أكبر في تنفيـذ البرامج .

2- استخدمت الذاكرات شبه الموصلة مما أدى إلى زيادة سعة الذاكـــرة وزيـــادة سرعة قراءة المعلومات من الذاكرة .

3- زيادة سرعة الأداء حيث أنها أصبحت تقاس بالنانوثانيـــة Nono second والتي تساوي إلى $1/(10)^9$ من الثانية .

4- استخدمت أنظمة التشغيل مما أدى إلى زيادة فعالية وأداء الأجهزة ، منها نظـام البرمجة المتعددة Multiprogramming .

5- ظهرت لغات برمجة جديدة لبرمجة الحواسيب .

6- استخدمت أجهزة إدخال وإخراج جديدة مثـــل أجـــهزة القـــراءة الضوئيـــة الشاشات الملونة .

7- ظهور حواسيب الميني المتوسطة الحجم .

رابعاً : حواسيب الجيل الرابع Forth Generation Computers

والذي يعتبر بدايته عام 1971 مع إنتاج أول معالج باسم (4004) مـــن شركة انتل Intel (المعالج هو عبارة عن شرائح إلكترونية متكاملة لعبـــت ومـــا زالت تلعب دوراً هاماً في إنتاج الحواسيب ، سنأتي على شرح المعالجات في فصـــول لاحقة).

من أهم ميزات هذا الجيل :

1. انخفاض سعرها بسبب إنتاج حواسيب الميكروكمبيوتر Micro computer أحجامها صغيرة وكلفته قليلة وذات إمكانيات هائلة .
2. ظهور المعالجات الميكروية Micro processors وميزالها الفريدة .
3. استمرار تطوير نظم التشغيل وظهور نظام تشغيل الأقراص Disk Operating System (DOS) للحواسيب الصغيرة .
4. ظهور وسائط التخزين (الأقراص المرنة) Floppy Disk وظهور الراسمات وشاشات العرض المرئي .
5. في مجال البرمجيات ظهور العديد من لغات البرمجة العالية المستوى منها الباسكال (Pascal) ولغة ال C والبرولوخ .Prolog .
6. ظهور الحزم البرمجية التطبيقية المتكاملة Integrated software packages ، والتي تتيح للمستخدم بمتابعة تطبيقات متعددة في آن واحد عن طريق استخدام النوافذ Windows.

وقد تعايش مع هذا الجيل انتشار شبكات الحاسوب على المستوى المحلي والعالمي وما تبع ذلك من تطور في نظم تشغيل الشبكات وتصاعد مشاكل المواجهة بين أنواع الحواسيب المختلفة ومشاكل أمن البيانات .

خامساً: حواسيب الجيل الخامس والأجيال المستقبلية

Fifth Generation Computers

يعتبر البعض أن هذا الجيل هو استمرارية للجيل الرابع ، والبعض يعتبر بداية التسعينات هو بداية جيل جديد من الحواسيب وذلك لظهور حواسيب متميزة ذات مواصفات هائلة وقد بدأ اليابانيون منذ منتصف الثمانينات في التخطيط لجيل الحواسيب الخامس التي تمتاز بالقدرة على فهم اللغات المحكية والمرسومة من خلال التعامل مع لغات قريبة من لغات الإنسان بالإضافة إلى سرعة معالجة كبيرة ، وفي عام 1992 تم إنتاج أول نموذج لهذا الجيل .

وقد أدى تطوير البرمجيات إلى ظهور النظام ويندوز Windows 3.0 عام 1990 الذي لم يكن نظام تشغيل قائم بحد ذاته فحسب ، بل شريحة برمجية تستند على النظام DOS . وفي عام 1991 طرحت شركة مايكروسوفت ويندوز3.1 ثم ويندوز3.11 (WINDOWS 3.11)، الذين وفرا دعما افضل لتكامل التطبيقات ووظائف السحب والإسقاط DRAG and DROP مما جعل هذا النظام مهيمنا على صناعة البرمجيات في مطلع التسعينات، بعد ذلك طور النظام ويندوز فظهر (Windows 95)، (Windows 97)، (Windows 98) مما أعطى هذا بعداً عملياً وفعالاً في مجال استخدامات الحاسوب في شتى المجالات .

أضف إلى ذلك فقد حققت لغتا فيجول بيسك Visual Basic وفيجول سي Visual c++من مايكروسوفت انتصارات هامة في مجال اللغات وتطبيقاتها .

أما في مجال العتاد فقد تم تطوير المعالجات بشكل ملحوظ فعلى سبيل المثال طرحت شركة INTEL عام 1993 المعالج بنتيوم Pentium الذي احتوى على حوالي 3 مليون ترانزستور ، وبموازاة تقدم المعالجات تطورت أيضاً تقنيات الأقراص الصلبة والمرنة وسعات تخزينها وكذلك تقنيات عرض الرسومات على الشاشة ، فظهرت المسرعات الرسومية لتحسين استجابة الشاشة مع تطبيقات ويندوز .

في مجال الأبحاث المتعلقة بتطوير الجيل الخامس والأجيال المستقبلية المتوقع إنتاجها ، فقد بدأت ، وما زالت مستمرة حتى وقتنا الراهن ، الأبحاث تتجه في اتجاهين في محاولة محاكاة العقل البشري :

الاتجاه الأول : يحاول تمثيل الحاسوب كشبكات عصبية وهو ما يعرف Artificial Neural Network .

الاتجاه الثاني : فيحاول من خلال التعاون مع علماء الهندسة الوراثية في إنتاج رقيقة حيوية وذلك بتكييف البروتينات لتحل محل السيليكون في

الدوائر الإلكترونية ، وتخدم أبحاث هذين الاتجاهين حواسيب الجيل السادس المتوقع إنتاجها .

أضف إلى ما سبق تطور أساليب الاتصالات وشبكات الحواسيب ونخص بالذكر شبكات الأنترنت Internet Network والتي سنتناولها في فصول لاحقة، والسؤال الذي يبقى مطروحاً : كم من هذه الأبحاث والمشاريع والمبادرات سيكتب لها النجاح ؟؟.

والجواب : من الصعب التنبؤ بالمستقبل ولكن من المؤكد أن تطور شبكات الأنترنت وشبكات الاتصالات الحاسوبية الأخرى ستلعب دوراً حاسماً في هذا الصدد، وكذلك التقدم المتزايد في تكنولوجيا العتاد والبرمجيات ، ومن المؤكد أيضاً أن السنوات القادمة من عمر الحواسيب الشخصية لن تكون أقل إثارة من تلك السنوات التي مضت ، وأن غداً لناظره قريب !

تمارين وأسئلة عامة

1. عدد مع الشرح العناصر الأساسية للنظام الحاسوبي ؟
2. عرف الحاسوب وبين علاقته بالإنسان ؟
3. تكلم بإيجاز عن أهم ميزات وإمكانيات الحاسوب ؟
4. هل يمكن للحاسوب أن يفكر بدلاً عن الإنسان ، ولماذا ؟
5. بين الفرق بين الحواسيب الرقمية والحواسيب القياسية ؟
6. اذكر أهم ميزات الحواسيب الميني ؟
7. عدد خمساً من استخدامات الحاسوب في الحياة ؟
8. اختر الجواب الصحيح من الأسئلة التالية :

1- وظيفة الحاسوب هي :

أ- معالجة البيانات . ب- تخزين البيانات .

ج- إخراج المعلومات . د - كل ما ذكر .

2- ظهرت الأقراص المرنة كوسيلة للتخزين في :

أ- الجيل الأول . ب- الجيل الثاني .

ج- الجيل الثالث . د- الجيل الرابع .

3. استخدمت لغة التجميع في برمجة الحواسيب بشكل واسع في :

أ- الجيل الأول . ب- الجيل الثاني .

ج- الجيل الثالث . د- الجيل الرابع .

4- يطلق اسم حواسيب الدارات المتكاملة على حواسيب الجيل :

أ - الأول . ب - الثاني .

ج - الثالث . د- الرابع .

5- ظـــهرت المعالجـــات الميكرويـــة Micro Processor مصاحبـــة للحواسيب:

أ- المتوسطة الحجم . ب- المصغرة .

ج- الكبيرة . د- العملاقة .

6- الأساس التكنولوجي للحاسوب العصري هو :

أ- الترانزستور . ب- الدارات المتكاملة .

ج- الصمامات المفرغة . د- المفتاح .

7- ظهرت الحواســيب المصغــرة (Micro computers) مصاحبـــة لتكنولوجيا :

أ- الصمامات المفرغة . ب- الدوائر المتكاملة .

ج- الترانزستور . د- أ + ب .

8. أستخدمت حواسيب الجيل الثاني :

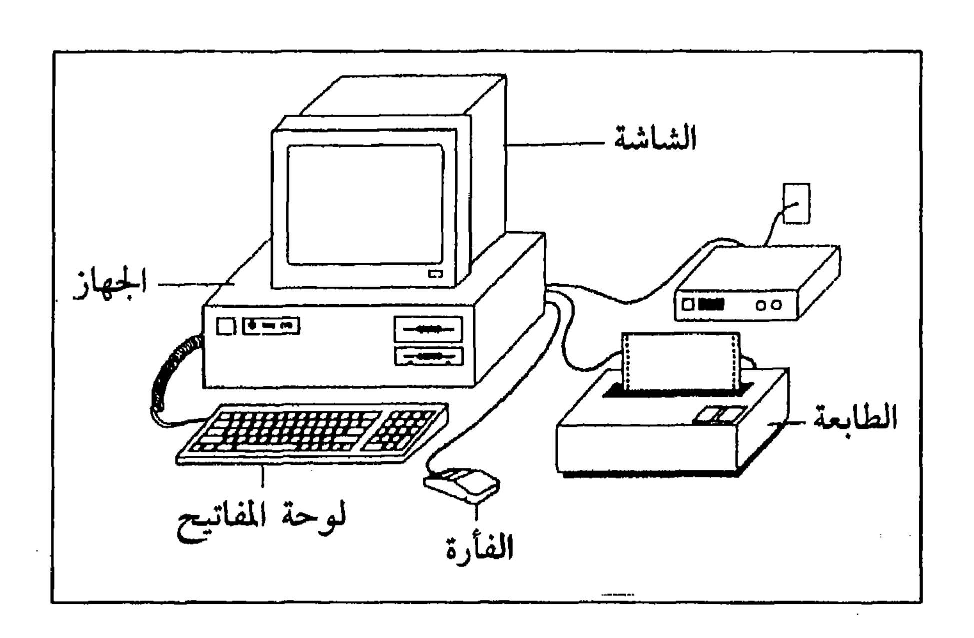
أ- ذاكرة الحلقات المغناطيسية . ب- ذاكرة شبه موصل .

ج- ترانزيستور . د- الدارات المتكاملة .

الفصل الثاني
مكونات نظام الحاسوب
Components of Computer System

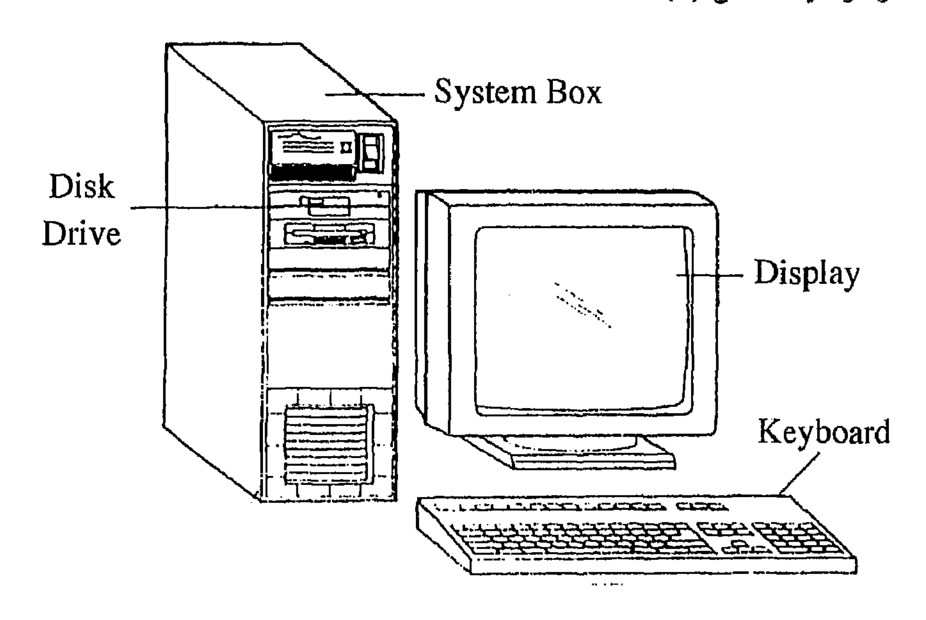
(2 – 1) مقدمـــة

يتكون أي حاسوب من العديد من الأجزاء التي تعمل سوية لانجاز العديـــد من المهام المختلفة ، كما في الشكل (1) ، سوف نقوم بشرح المكونات و آليـــة عملها بشكل مفصل.



الشكل (1) المكونات الاساسية لجهاز الحاسب

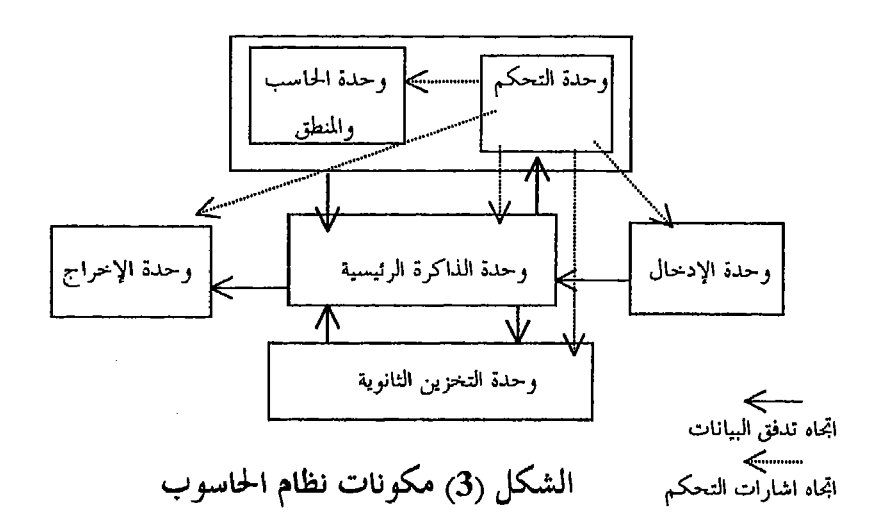
هناك نماذج أخرى للحاسبات الشخصية الشائعة الاستعمال منها النمـــوذج الموجود في الشكل (2) :



الشكل (2) أحد نماذج الحاسبات الشخصية الشائعة

إن أي حاسوب يتكون من الأجزاء الأساسية التالية كما هـــو واضـــح في الشكل (3) :

1 - وحدة التشغيل المركزية	Central Processing Unit (CPU)
2 - وحدة الذاكرة الرئيسية	Main Memory
3 - وحدات التخزين الثانوية	Secondary Storage
4 - وحدات الإدخال	Input Units
5 - وحدات الإخراج	Output Unit



الشكل (3) مكونات نظام الحاسوب

(2 – 2) وحدة المعالجة المركزية

Central Processing Unit (CPU)

يطلق على وحدة الحساب و المنطق و وحدة التحكم معاً إسم وحدة المعالجة المركزية أو المعالج Processor، والمعالج في الحاسوب كالدماغ في الانسان فهو الذي يفكر ويحلل المعلومات ويتخذ القرارات ، إلى ما هنالك من أعمال أخرى . للمعالج عدة أنواع ، أشهرها معالجات الحواسيب الشخصية من انتاج شركة INTEL ، وسنبين فيما يلي أهم المعالجات و سرعتها وفي أي الحواسيب الشخصية استخدمت :

نوع المعالج CPU	سرعة المعالج Speed	حجم ذاكرة المعالج Memory	نوع الحاسوب Computer
80.8085	0.4 MHZ	256 KB	Pc
8086.8088	4-10 MHZ	640 KB	XT
80286	12-20 MHZ	1 MB	AT
80486	16-40 MHZ	1 MB	AT 386
80486	23-100 MHZ	4 MB	AT 486
80586 (Pentium)	60-400 MHZ	8,16,32,64, 128,256 MB	Pentium

من الجدير بالذكرأن للمعالج Pentium عدة أنواع حسب ا لسرعة ، منها :بنتيوم 75 ، بنتيوم 100 ، بنتيوم 133 ، بنتيوم 166 ، بنتيوم 200 ، بنتيوم 266 ، وبنتيوم 300 ، وبنتيوم 400 وبنتيوم 450 .

تعتبر وحدة المعالجة المركزية من الوحدات الأساسية في الحاسوب فهي تمثل القلب النابض للحاسوب لأنها تحتوي على جميع الامكانيات الضرورية لتداول البيانات والتعليمات و ضبط جميع العمليات من حساب وتخزين و استرجاع .كما تعتبر من الوحدات الأساسية التي تحدد قدرة و سرعة وثمن الحاسوب .

2 – 2 – 1) وحدة الحساب والمنطق

Arithmetical and Logical Unit

تتكون وحدة الحساب و المنطق من :

أ. مجموعة من الدوائر الالكترونية المنطقية التي يتم توظيفها لأداء العمليات التي تتم داخل ذاكرة الحاسوب .

ب. مجموعة من المسجلات Registers، تستخدم هذه الوحدة لاستيعاب البيانات التي يجري معالجتها ، و كذلك استيعاب النتائج الوسيطة اثناء عملية المعالجة ، ويمكن التمييز بين أنواع المسجلات التالية التي تضمها وحدة الحساب و المنطق :

1. مسجلات النقطة العائمة Floating Point Registers والتي تستخدم لانجاز العمليات الحسابية على الارقام الحسابية ذات النقطة العائمة والتي سيتم شرحها لاحقاً .
2. مسجلات النقطة الثابتة Fixed Point Registers و التي تستخدم لانجاز العمليات الحسابية على الارقام الحسابية ذات النقطة الثابتة .

3. المسجل المركم Accumulator وهو مسجل خاص يستخدم لتجميع نتائج العمليات الحسابية التي تنجزها هذه الوحدة ويعتبر المراكم هو المسجل الرئيسي في وحدة الحساب والمنطق ، ويدعى أيضاً المسجل .

4. مسجلات الازاحة Shifting Registers تستخدم في انجاز عمليات الازاحة .

5. عدد من المسجلات عامة الغرض General Purpose Registers تساعد في انجاز المهام المنوطة بهذه الوحدة ، و عادة ما تعنون هذه المسجلات بعنوان رقمي يبدأ بالرقم ...,0,1,2,3,4 .

وظيفة وحدة الحساب والمنطق :

وحدة الحساب والمنطق مسؤولة عن انجاز ما يلي :

- كافة العمليات الحسابية مثل الجمع والطرح والقسمة والضرب والرفع للأس ..
- كافة العمليات المنطقية الأساسية OR , AND , NOT.
- عمليات المقارنة وتشمل أكبر من ، و أصغر من ، أصغر أو يساوي ، أكبر أو يساوي ، يساوي ، لا يساوي .
- عمليات الإزاحة SHIFT والمقصود بها تحريك محتويات الوحدة التخزينية إلى اليمين أو اليسار لأغراض المعالجة مع الاخذ بعين الاعتبار إن الإزاحة خانة واحدة إلى اليسار تعني ضرب الرقم ب 2 و الإزاحة إلى اليمين تعني القسمة على 2 (وهو أساس النظام الثنائي).

مثال يوضح آلية عمل وحدة الحساب والمنطق :

لتكن لدينا المعادلة التالية :

$$Z = X + Y$$

ولنرى كيف يتم انجاز عملية الحساب في هذه المعادلة .

يتم تخزين قيمة X وقيمة Y أولا في الذاكرة ومن ثم يتم انجاز عملية الجمع وفقاً للتتابع التالي:

1- يتم نقل قيمة X إلى وحدة الحساب والمنطق حيث يتم تخزينها في المسجل A.

2- تقوم الدوائر المنطقية بجلب قيمة Y وإضافتها إلى قيمة المراكم .

3- يتم تخزين ناتج الجمع في المراكم .

4- يتم نقل محتويات المراكم إلى المخزن Y في الذاكرة .

واضح من التتابع السابق أن المركم نفسه لا يقوم بالعملية الحسابية ولكنه يستخدم في تجميع نتائج العمليات تمهيداً لتخزينها في الذاكرة ، يكون طول المراكم مساوياً لطول الوعاء التخزيني في الذاكرة ، لذلك فإنه في العمليات الحسابية التي يحتاج نتائجها الوسيطية إلى وعاء تخزيني اكبر مثل عمليات الضرب يتم الحاق مسجل اضافي بالمراكم ليصبح في هذه الحالة مركماً مضاعف السعة .

(2-2-2) وحدة التحكم : CONTROL UNIT

اذا كانت وحدة المعالجة المركزية هي قلب الحاسوب فان وحدة التحكم هي قلب وحدة المعالجة المركزية فهي التي تقوم بعملية التنسيق بين أعمال وحدات الحاسوب الأخرى بغرض تنفيذ البرنامج المخزن مسبقاً في الذاكرة وبالتالي فإن دورها يبدأ في بداية التشغيل ولا ينتهي إلا بانتهائه .

مكونات وحدة التحكم :

تتكون وحدة التحكم من العناصر التالية :

1- عناصر توقيت الأعمال التي يؤديها الحاسوب وتقوم بالمهام التالية :

- ضبط التوقيت لتسلسل تعليمات البرنامج .
- نقل التعليمة التالية في التنفيذ من موقعه في الذاكرة إلى وحدة المعالجة المركزية.
- تحليل التعليمة وتحديد العملية المراد تنفيذها .

- اصدار أوامر للتحكم إلى باقي وحدات الحاسوب المعنيــة في التنفيــذ علــى شكل اشارات تحكم وتطلب منها اداء العملية المراد تنفيذها .

2-عناصر متابعة وتنفيذ الأوامر الحاسوبية وهي عبارة عــن مجموعــة الــدارات الالكترونية التي تتحد مع بعضها لتكون ما يسمى بالمسجلات ، أهــم هــذه المسجلات :

- مسجل الأمر الحالي Instruction Register وباختصار (IR) وهــو الذي يستخدم في تخزين التعليمة التي يجري تنفيذها بعد جلبها من الذاكرة
- عداد البرنامج Program Counter وباختصار (PC) وهو عبارة عــن مسجل يحتفظ بعنوان التعليمية التالية في التنفيذ ، ويقوم بتعداد الأوامــر الــتي يجري تنفيذها من خلال تحديده للامر التالي في التنفيذ، ولتوضيح ذلك نفترض اننا بصدد تشغيل برنامج موجود فعلا في الذاكرة وكانت محتويــات العــداد صفرا ، فان ذلك يعني ان الأمر التالي في التنفيذ هو الأمر الذي عنوانه صفــر ، يزداد العداد بالقيمة 1 فيصبح الأمر التالي لتنفيذ هو 1 ...الخ
- محلل التعليمه Instruction Decoder باختصار (ID) وهو عبارة عــن مسجل يعمل على تحليل شيفرة التعليمة بهدف تحديد نــوع العمليــة المــراد تنفيذها وهذا المسجل هو عبارة عن دائرة إلكترونيه لها مدخل وعــدد مــن المخارج مساوٍ لعدد العمليات التي يمكن لوحــدة المعالجــة المركزيــة القيــام بتنفيذها.

وظائف وحدة التحكم :

إن وحدة التحكم كما قلنا هي التي تقوم بالتنسيق بين وحدات الحاســوب الأخرى و ضبط كافة العمليات التي تتم داخل وحدة المعالجة المركزيــة ويمكــن بالإضافة إلى ذلك تحديد مهام هذه الوحدة في الآتي :

1. يتم ادخال البيانات و التعليمات إلى ذاكرة الحاسوب بناءاً علــى إشــارات

خاصة من وحدة التحكم .

2. تقوم وحدة التحكم بجلب الأمر التالي في التنفيذ من موقعه في الذاكرة و إعادة تخزينه في المسجل المعد لهذا الغرض .

3. اصدار إشارات إلى كل من وحدة الحاسب و المنطق وكذلك الذاكــرة مــن أجل البدء بتنفيذ الأمر ، وإعادة تخزين النتائج في الذاكرة تمهيداً لإخراجـــها على إحدى وسائط الإخراج .

4. تحديد عنوان الأمر التالي في التنفيذ .

5. يتم تكرار هذه العمليات حتى ينتهي البرنامج .

دورة المعالجة في وحدة التحكم :

Processing Cycle in control unit :

يتم تنفيذ التعليمات والأوامر داخل الحاسوب في دورتين رئيسيتين لوحـدة التحكم هما :

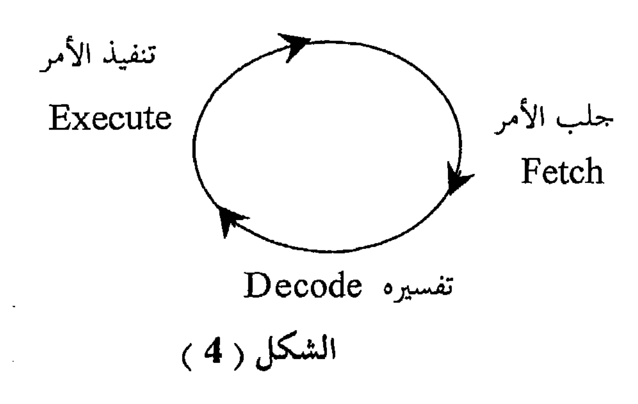
أ. دورة التعلّيمة Instruction Cycle :أي تحديد الأمـــر و التعليمـــة المـــراد معالجتها ، ويمكن إيجازها كما يلي :

1. جلب Fetch الأمر أو التعليمة من الذاكرة و تخزين عنوانهـــا في المســـجل (PC) وتخزينها في المسجل (IR) .

2. تفسير التعليمة و فك شيفرتها Decoding و تحديد نوع العمليـــة المـــراد تنفيذها .

3. اصدار الاشارات اللازمة لتنفيذ هذه التعليمة و لمعرفة التعليمـــة التاليـــة في التنفيذ.

4. تكرار الخطوات السابقة حتى تنتهي الأوامر و التعليمـــات ، الشـــكل (4) يلخص هذه الدورة .



الشكل (4)

ب. دورة التنفيذ Execution Cycle :

حيث يتم في هذه الدورة جلب البيانات طبقـــاً للأوامـــر أو التعليمـــات بواسطة وحدة التحكم، إلى وحدة الحساب والمنطق حيث تتم معالجتـــها ومـــن ثم إعادة تخزين النتائج المرحلية في مسجلات معينة وإعادة تخزين النتائج النهائيـــة في ذاكرة الحاسوب الرئيسية.

أولويات تنفيذ العمليات الحسابية داخل وحدة المعالجة الرئيسية:

تقوم وحدة الحساب والمنطق بانجاز العمليات على التعابير الحسابية وفـــق ترتيب محدد من قبل البرمجيات ، و يتم تنفيذه تحت اشراف وحدة التحكم علـــى الشكل التالي :

1. الرفع الى أس .
2. الضرب و القسمة .
3. الجمع و الطرح .

إذا استخدمت الأقواس في التعابير الحسابية فان ما بين القوسين يتم تنفيـــذه أولاً وحسب قاعدة الأولويات ، وفي حال وجود أقواس متداخلة يتم تنفيذ القـــوس الداخلي أولاً ثم الخارجي .

مثال 1 : يتم تنفيذ التعبير الحسابي التالي :

$$A - B + 8 / 3 * C ** 4$$

كما يلي :

أ- يتم تنفيذ 4 ** C أولا .

ب- في الخطوه الثانية يتم تنفيذ 3 / 8 .

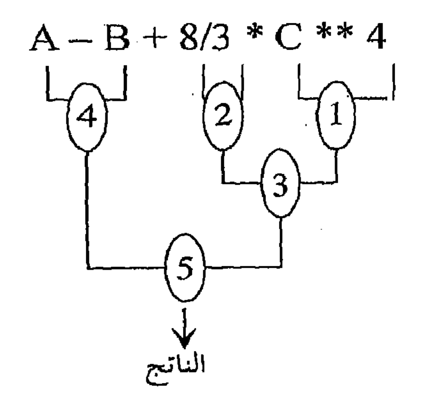
ت- في الخطـــوة الثالثـــة يتـــم تنفيـــذ 4 ** C * 8/3 أي ضــــــرب ناتج (أ) بنـــــاتج (ب).

ث- في الخطوة الرابعة يتم تنفيذ A - B .

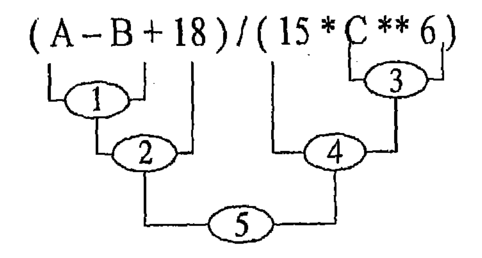
ج- في الخطوة الخامسة يتم تنفيذ

$$A - B + 8 / 3 * C ** 4$$

أي جمع ناتج (ث) مع ناتج (ت) .



مثال 2 : يتم تنفيذ التعبير الحسابي التالي :



كما يلي :

أ. يتم تنفيذ A – B داخل القوس الأول .

ب. يتم تنفيذ العملية الحسابية داخل القوس الأول

A – B + 18

ت. يتم تنفيذ عملية الرفع إلى اسس داخل القوس الثاني

C ** 6

ث. يتم تنفيذ العملية الحسابية داخل القوس الثاني

15 * C ** 6

ج. الخطوة الاخيرة يتم تنفيذ عملية القسمة .

(A – B + 18) / (15 * C ** 6)

(2 – 3) وحدات التخزين الداخلية Internal storage Units

تتعدد وحدات التخزين الداخلية وتتنوع في الشكل والسعة والوظيفة ، ويمكن التمييز بين الأنواع التالية :

1- الذاكرة المؤقتة Random Access Memory (RAM) .

2- الذاكرة الدائمة Read Only Memory (ROM) .

3- الذاكرة الفورية Cache Memory .

4- المسجلات Registers .

5- المخازن الانتقالية Buffers .

(2–3–1) الذاكرة الرئيسية Main Memory

وهي ذاكرة القراءة والكتابة ، حيث يشار اليها بمسميات متعددة منها :

أ – المخزن الاساس Primary Storage .

ب – المخزن الرئيسي Main Storage .

ج- ذاكرة الوصول العشوائي Random Access Memory .

عناصر بناء الذاكرة الرئيسية :

هناك العديد من العناصر التي استخدمت لبناء الذاكرة الرئيسية للحاسوب، فقد تم استخدام الصمامات في أول حاسوب ، كما تم استخدام الاسطوانات الممغنطة في منتصف الخمسينات ، كما استخدمت الحلقات الممغنطة خلال الفترة الواقعة في بداية الستينات إلى منتصف السبعينات منذ منتصف السبعينات بدء استخدام أنصاف النواقل لبناء الذاكرة .

إن الذاكرة المصنوعة من أنصاف النواقل تفقد كل ما هو مخزن بها عند فصل التيار الكهربائي عن الحاسوب ، ولهذا السبب فإن أنظمة الحاسبات الكبيرة التي تستخدم هذا النوع من الذواكر تستخدم نظام تغذية احتياطي Uninterruptible Power Supply واختصاراً (UPS) وذلك من أجل الحفاظ على محتويات الذاكرة الرئيسية لفترة زمنية أطول في حالة الانقطاع المفاجئ للتيار الكهربائي .

الذاكرة الرئيسية هي ذاكرة القراءة والكتابة :

تتكون الذاكرة الرئيسية من عدد من المواقع التخزينية ، لو فرضنا أن عددها N فإنها تكون مرقمة من 0 إلى N – 1 ، يدل على عنوان الموقع ، يستخدم العنوان للوصول إلى الموقع التخزيني ، وعندما يتم تخزين أي عنصر جديد من البيانات أو التعليمات في موقع تخزين فان المحتويات السابقة في هذا الموقع يتم مسحها وازالتها وحفظ ما هو جديد بدلاً منها ، ويطلق على عملية ادخال بيانات أو تعليمات جديدة إلى الذاكرة عملية الكتابة (Writing) ، وعندما يتم استرجاع البيانات والتعليمات من موقع التخزين بالذاكرة الرئيسية فإنه يتم قراءتها (Reading) ثم عمل نسخة منها في مكان آخر ولا يتم إزالتها من موقع التخزين الرئيسي .

يمكن تشبيه عملية القراءة بقراءة الإنسان لكتاب ما ، فهو يقرأ منه وقت مـا يشاء ولكنه لا يغير موقع الكتابة فيه ، أما عملية الكتابة فيمكن تشـبيهها بعمليـة كتابة الإنسان في الكراسة حيث يستطيع الكتابة فيها ثم يقرأ منها ما كتبه كمـا يمكننا مسح أي جزء ثم كتابته وإضافة جزء جديد بدلاً منه ونظراً لإمكانية الكتابـة على الذاكرة الرئيسية وإمكانية القراءة منها فإنه يطلق عليها أسم ذاكـرة القـراءة والكتابة .

كل موقع في الذاكرة يتكون من عدد متساوٍ من الخلايا يخصص كل منـها لحفظ الرقم "0" أو الرقم "1" ، فتسمى الخلية الواحدة بثنائية bit (وهي أصغـر وحدة تخزين تقاس بها سعة الحاسوب) ، لذلك فأن حجم الموقـع يحـدد بعـدد الثنائيات التي يتكون منها ، وهذا يختلف من حاسوب لأخر .

عادة يتكون الموقع التخزيني المرتبط بعنوان من 8 ثنائيـات (أي bit 8) والذي يدعى نظام ترميز معين نذكر منها شيفرة الاسـكي ASCII وشـيفرة الايبسدك EBCDIC .

ملاحظة هامة

- أن وحدة التخزين هي الثنائية Bit .
- لتمثيل رمز وحرف في الحاسوب يستخدم ما يسمى البايت Byte

1 byte = 8 bit

- لتخزين عدد او جمل ذات معنى تستخدم ما يسمى Word وحيث ان حجـم الـ Word يتراوح من 8 ثنائيات إلى 64 ثنائية ففي الحواسـيب الصغـيرة يكـون حجـم الكلمـة 8 ثنائيـات أي Byte

و في الحواسيب الأكبر يكون حجم الكلمة يساوي 16Bit ويساوي 2Byte

و في الحواسيب الأكبر يكون حجم الكلمة يساوي 32Bit ويساوي 4Byte

و في الحواسيب الأكبر يكون حجم الكلمة يساوي 64Bit ويساوي 8Byte

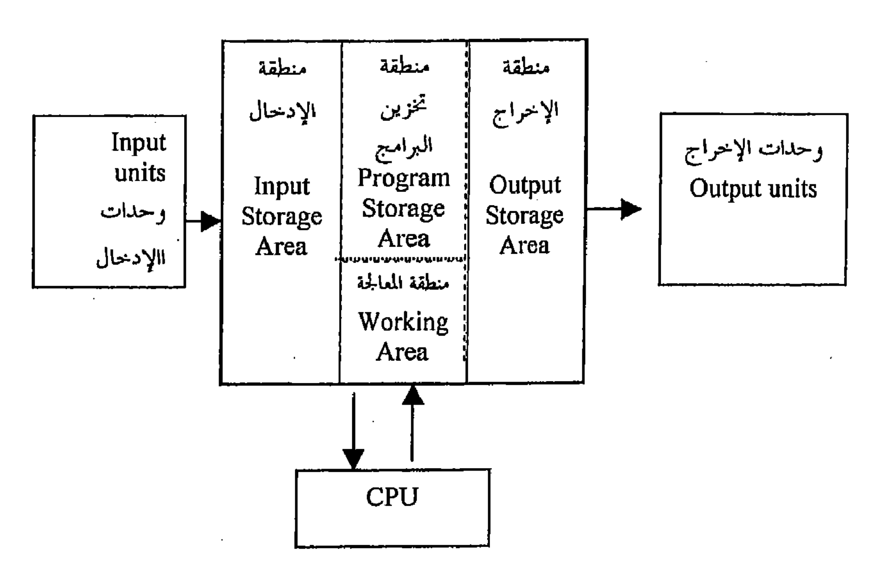
ولهذا فإن Word تتكون من 4Byte وكل 1Byte = 8 Bits

- يعتبر حجم الكلمة إحدى الصفات الرئيسية للحاسوب ، حيث ترتبط وحـــدة الذاكرة الرئيسية بوحدة المعالجة المركزية بواسطة ناقلات Buss ، يـــتم مـــن خلالها تبادل المعلومات ، فكلما كانت حجم الكلمة اكبر كلما كانت عمليـــة نقل البيانات بين وحدة الذاكرة ووحدة المعالجة المركزية تتم بشكل اسرع .

- الناقل Buss عبارة عن أسلاك تربط وحدة المعالجة المركزية ووحدة الذاكـــرة وكذلك لربط الذاكرة مع أجهزة الإدخال و الإخـــــراج ووحـــدة التخزيـــن الثانوية.

مناطق الذاكرة :

تنقسم الذاكرة الرئيسية إلى مناطق ، كما هو موضح بالشـــــكل (5) حيث لا تفصل هذه المناطق أي فواصل أو حدود ماديـــة ، فـــالخطوط المتقطعـــة بالشكل (5) ليست فواصل طبيعية ولكنها خطوط وهمية ، اذ ان حجــــم كــــل منطقة من هذه المناطق يختلف من برنامج إلى آخر و هذه المناطق هي :



الشكل (5) يمثل مناطق تقسيم الذاكرة الرئيسية

1- منطقة البرامج Program Area :

وهو مخصص لتخزين البرامج والأوامر والتعليمات سواء تلك المتعلقة بنظـــام التشغيل أو البرامج التطبيقية .

2- منطقة المدخلات Input Area :

مخصص لاستقبال البيانات من وحدات الإدخال وتسمى هذه المنطقـة Input buffer حيث Buffer هنا هي وحدات تخزينية مخصصة لاستقبال البيانـات النشطة وتخزينها مؤقتاً إلى أن يتم معالجتها .

3- منطقة المخرجات Output area :

هذه المنطقة مخصصة لاستقبال النتائج بعد معالجتها وتمهيداً لاخراجها وتسـمى هذه المنطقة ايضاً Output Buffer إذا أن التخزين فيها يكون تخزينا مؤقتـا للمعلومات النشطة تمهيداً لاخراجها بواسطة وحدة الإخراج المناسبة .

4- منطقة العمل (المعالجة) Working area :

وتعمل هذه المنطقة بمثابة هامش أو مسودة حيث يتم فيـها تخزيـن النتـائج الوسيطية والبيانات المفترضة التي لا تعرف ضمن المدخلات .

معايير التميز بين الذاكرات :

من أهم العوامل الاساسية التي يمكن ان تتميز بها الذاكرة ما يلي :

1- السعة Capacity :

ويقصد بها عدد المواقع التخزينية في الذاكرة ، و معظم الحواسيب الصغيرة تقـاس سعة ذاكرتها بالبايت أي : 1 Byte = 8 Bit.

و في بعض الاخر تقاس بالكيلوبايت Kilo Byte حيث :

$$1\ K.B = 2^{10}\ Byte = 1024\ Byte$$

و بعض الأجهزة تقاس بالميغابايت Mega Byte حيث :

$$1\ M.B = 2^{10}\ K.B = 2^{20}\ Byte$$

واحيانا تقاس بالجيجا بايت Giga Byte حيث :

$$1\ G.B = 2^{10}\ M.B = 2^{20}\ K.B\ = 2^{30}\ Byte$$

وفي الوقت الراهن أصبح يستخدم مقياس تيرا بايت Tera Byte حيث :

$$1\ T.B = 2^{10}\ G.B\ = 2^{40}\ Byte$$

2- زمن الوصول Access Time :

وهو الزمن الذي يستغرقه الحاسوب في قراءة كلمة واحدة (Word) من الذاكرة ،كلما زادت سرعة الوصول زادت سرعة الحاسوب في تنفيذ تعليمات البرنامج المعالج وبالتالي زاد مستوى أداء الحاسوب .

3- طريقة الوصول Access Method :

يقصد بها كيفية الوصول إلى المعلومات المخزنة ويتم ذلك بطريقتين :

أ- طريقة الوصول المباشر أو العشوائي : Random Access Method

أي الوصول إلى المعلومات المطلوبة مباشرة دون الحاجة إلى المرور على ما يسبقها من بيانات .

ب-طريقة الوصول غير المباشر أو التتابعي S eguential Access Method

أي الوصول إلى المعلومات المطلوبة بشكل تسلسلي (تتابعي) بعد المرور على معلومات جميع المواقع السابقة للموقع المطلوب .

(2- 3- 2) الذاكرة الفورية Cache Memory

وتسمى أيضا بالذاكرة المسرعة وتستخدم خلال عملية التشغيل وهي عبارة عن ذاكرة تخزين مؤقت ذات سرعة عالية جداً تفوق سرعة الذاكرة RAM بكثير. وهي تستخدم للتخزين المؤقت للبيانات والتعليمات المطلوب استرجاعها مرات عديدة أثناء عملية معالجة البيانات مما يساعد على سرعة معالجة البيانات

(2 – 3 - 3) الذاكرة الدائمة Read Only Memory ROM

وتسمى بذاكرة القراءة فقط ، حيث أن استخدامها يقتصر على القراءة فقط ومن مهامها الرئيسية الإشراف على بدء تشغيل الحاسوب . من خصائص هذه الذاكرة :

1. تحتوي على البرامج والروتينات التي تحتاج اليها بصفة دائمة ، مثل :

- مترجم (مفسر) اللغة الرئيسية للجهاز كمفسر لغة بيسك في معظم الأجهزة المصغرة .
- روتينات الإدخال و الإخراج (Input and Output Routines)
- روتينات بدء تشغيل الجهاز (Start-up Routines)

2. يمكن القراءة من هذه الذاكرة فقط وذلك عند الحاجة لأي من البرامج أو الروتينات السابقة ، ولا يمكن لمستخدم الجهاز أن يسجل في الذاكرة أية معلومات (يقرأ منها ولا يكتب فيها) .
3. لا تفقد محتوياتها سواء بعد القراءة منها أو عند فصل التيار الكهربائي .
4. محتويات هذه الذاكرة لا تتغير مطلقا من قبل المستخدمين لها .

أهم أشكال ذاكرة ROM :

1. ذاكرة ROM : هي ذاكرة مبرمجة لمرة واحدة ، أي يخزن فيها برنامج ثابت أو معلومات ثابته ولا يمكن اجراء أي تعديل على مضمونها .
2. ذاكرة القراءة القابلة للبرمجة

Programmable Read Only Memory (PROM)

بمجرد برمجة هذه الذاكرة فإنه لا يمكن تغيير ما هو مخزن بها ، أي أن الذاكرة PROM تصبح ROM وما هو مخزن عليها يمكن قراءته فقط .

3. ذاكرة القراءة القابلة للبرمجة و المسح (EPROM)

Erasable and Programmable Read Only Memory

كما هو واضح من التسمية فان ما تحمله هذه الذاكرة يمكن مسحه وإعادة برمجته من جديد لتسجيل تعليمات جديدة باستخدام وسائل خاصة للبرمجة وتتم عملية المسح بتعريض شريحة EPROM للاشعة فوق البنفسجية ، وهذه الذاكرة مثل الذاكرة ROM يمكن قراءة ما هو مسجل فيها فقط وتبقى هذه الذاكرة إلى أن يتم مسحها بالطريقة التي سبق ذكرها .

(2 –3– 4) المسجلات Registers :

تطلق تسمية مسجلات على الوحدات التخزينية الموجـــودة في وحـــدتي التحكم والحساب والمنطق ، ويمكن تعريف المسجل على انه : وحـــدة الكترونيـــة قادرة على استيعاب وحدة من المعلومات اثناء عملية المعالجة .

تعتبر المسجلات وحدات تخزينية مؤقته بمعنى أن المعلومات تخزن فيها لفـــترة وجيزة حتى تتم معالجتها ، كما أنها تتميز بسرعتها الفائقـــة في نقـــل المعلومـــات وتداولها بين وحدة الحساب والمنطق ووحدة التحكم والذاكرة .

عادة يكون طول المسجل مساويا لطول الكلمـــة Word في الذاكـــرة ، وهذا منطقي إذ أن المسجل يستخدم للتخزين المؤقت للبيانات والتعليمـــات الـــتي تنتقل اليه من الذاكرة أو تنتقل منه اليها . ويختلف عـــدد المســـجلات وأطوالهـــا وأنواعها من نظام إلى آخر وفيما يلي بعض أنواع المسجلات ووظيفة كل منها:

1- المراكم Accumulatos : وقد يسمى A – Register ، يستخدم في تجميع النتائج الحسابية الوسيطة ، وهو الرئيسي في وحدة الحساب والمنطق.

2- مسجل العنوان Address Register, : وهو يستخدم لتخزين عنـــوان أو رقم الموقع التخزيني بالذاكرة الذي يراد التعامل معه ، Memory Address Register ومسجل محتوى الذاكرة .

3- مسجلات وحدة التحكم : مثل مسجل التعليمـــة Instruction Register. ومسجل عداد البرامج Program Counter Register ومسجل تحليـــل التعليمة Instruction Decoder .

4- مسجلات عامة الغـــرض General Purpose Register : وتســـتخدم لتخزين أنواع مختلفة من البيانات .

5- مسجلات القراءة و الكتابة في الذاكرة RAM وهذه المسجلات هي :

أ. مسجل محتويات الذاكرة : Memory Buffer Register (MBR)

ب.مسجل عنوان الذاكرة : Memory Address Register (MAR)

(2 – 3 – 5) المخازن الانتقالية Buffers :

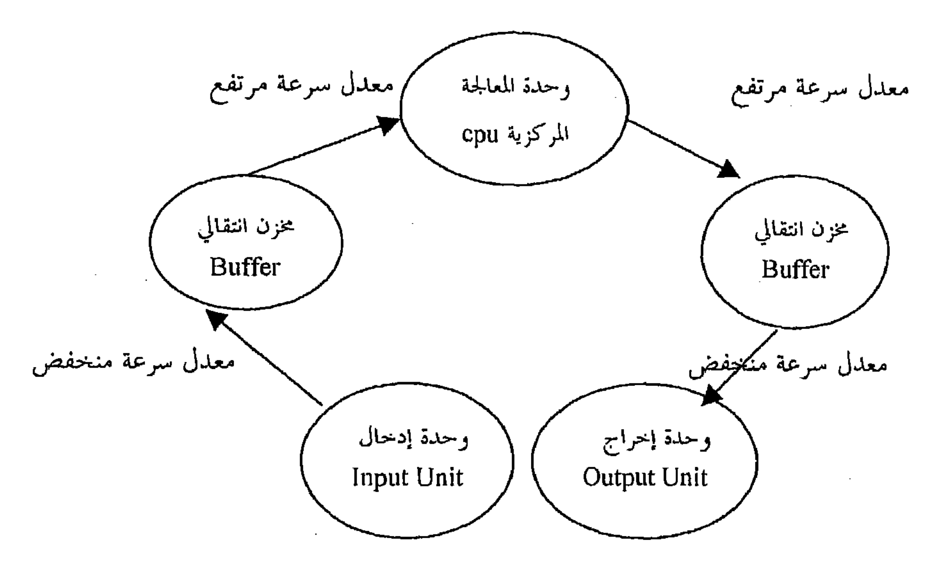
المخزن الانتقالي هو مساحة تخزينية بالذاكرة أو خارجها ذات سعة محدودة وسرعة تداوله عالية ، تستخدم لاستيعاب البيانات استيعاباً مؤقتاً تمهيداً لنقلها وتداولها بين أي وحدتين من وحدات الحاسوب . وتسمى هذه العملية بعملية التخزين الانتقالي طبقا لهذا المفهوم فان المسجلات بوحدة المعالجة المركزية تعتبر بصورة أو بأخرى مخازن انتقالية ، ذلك لأنها تستخدم في الاحتفاظ بالبيانات والتعليمات احتفاظاً مؤقتاً تمهيداً لتداولها بين وحدات التحكم ، الذاكرة، والحساب والمنطق إلا انه جرى العرف الاصطلاحي على تسميتها مسجل ، أما اصطلاح Buffers فانه يطلق على النوعين التاليين من المخازن الانتقالية :

أ- المخازن الانتقالية الملحقة بوحدات الإدخال والإخراج :

تستخدم هذه المخازن لتقليل من اثر فارق السرعة بين وحدات الإدخال والإخراج من ناحية ووحدة المعالجة المركزية من ناحية أخرى . فمثلاً عند إدخال بيانات عن طريق لوحة المفاتيح فإنه مهما بلغت سرعة مدخل البيانات فإنها لا تتعدى حرفاً أوحرفين في الثانية الواحدة وهذه السرعة أقل بكثير من سرعة وحدة المعالجة المركزية ، حيث يتم استقبال البيانات وتجميعها في كتل من ا لبيانات (سطور مثلاً) ومن ثم نقلها من هذا المخزن الانتقالي إلى وحدة المعالجة المركزية كتلة بكتلة وبسرعة عالية تتناسب وسرعة وحدة المعالجة المركزية .

كذلك الحال عند إخراج المعلومات من وحدة المعالجة المركزية ، إذ أن سرعة وحدة الطباعة مثلاً تكون بطيئة جداً بالنسبة لسرعة المعالج المركزي مما يؤدي إلى تعطيل وقت المعالج نتيجة لتأخر وحدة الطباعة وعدم قدرتها على مجاراة المعالج في سرعة نقل البيانات واخراجها . لذلك يستخدم مخزن إنتقالي يقع بين وحدة الإخراج ووحدة المعالجة المركزية ، حيث تنقل البيانات إليه بسرعة عالية هي سرعة

التداول بوحدة المعالجة المركزية التي تتفرغ لتنفيذ عملية أخرى ويتـــــولى المخـــزن الانتقالي نقل النتائج إلى وحدة الطباعة حسب سرعتها والشكل (6) يبين ذلك .



الشكل (6) يمثل المخازن الانتقالية الملحقة بوحدات الإدخال والإخراج

ب- المخازن الانتقالية بالذاكرة الرئيسية :

عند نقل البيانات من وحدة التخزين الخارجية مثل الاقراص والأشــــــرطة الممغنطة إلى وحدة المعالجة المركزية أو العكس ، تنشأ أيضاً مشكلة فارق الســــرعة بينها وللتغلب على هذه المشكلة يتم تخصيص قطاعات بالذاكرة الرئيسية تســتخدم كمخازن انتقالية .

فالبيانات في الملف تكون منظمة في وحدات بيانات تســـمى ســـجلات Records ، ويتم عادة تجميع عدد من السجلات لتعامل كوحدة واحدة تســـمى كتلة .

عند ادخال البيانات إلى الحاسوب من الملف يتم نقل كتلة كاملـــــة مـــن البيانات إلى المخزن الانتقالي بالذاكرة الذي يسمى في هذه الحالة مخزن المدخـــلات الانتقالي (Input Buffer) ، حيث تقوم وحدة المعالجة المركزية بالتعـــامل مـــع محتويات هذا المخزن الانتقالي سجلا بعد سجل .

كذلك عند إخراج المعلومات من الحاسوب إلى الملف ، فإنه يتم تجميع سجلات المعلومات على شكل كتل بمخزن انتقالي بالذاكرة الذي يسمى في هذه الحالة مخزن المخرجات الانتقالي (Output Buffer) ، ومن ثم يتم نقل محتوياته إلى الملف بوحدة التخزين الخارجي .

(2 – 4) وحدات التخزين الثانوية

Secondary Storage Units

يسمى التخزين الذي يتم في الذاكرة الداخلية للحاسوب بالتخزين الأولي أو التخزين الداخلي . كما يسمى التخزين الذي يتم في أماكن التخزين الأخرى بالتخزين الثانوي أو الخارجي (External Storage) ، كما يطلق في كثير من الاحيان على وسائل التخزين الثانوية وسائل التخزين المساعدة (Auxiliary Storage) .

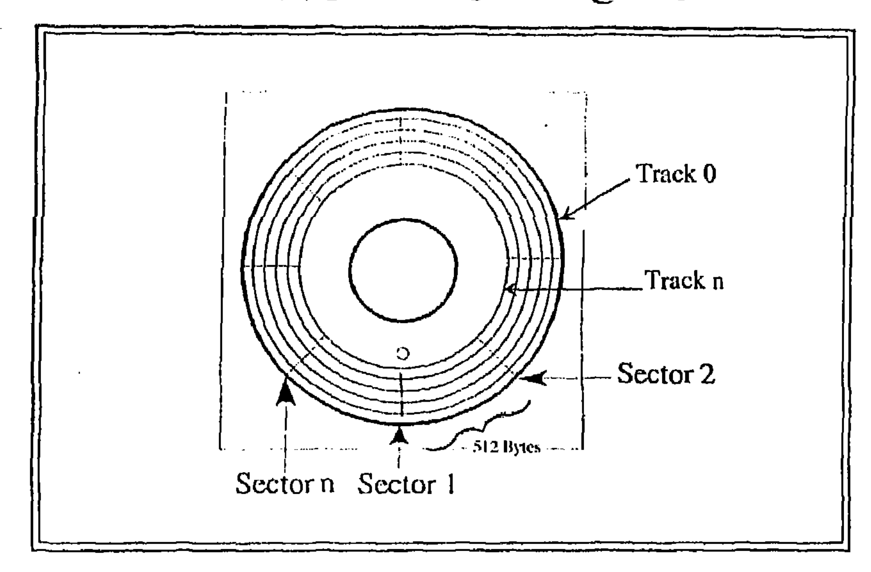
تعتبر الأقراص المغنطة (Magnetic Disks) أكثر وسائط التخزين شيوعاً واستخداماً ، وتتميز بإمكانية تخزين واسترجاع البيانات منها بطريقة مباشرة. من الناحية الشكلية يمكن تشبيه القرص بالاسطوانات التي تستخدم في تسجيل الاغاني والموسيقى، هناك اشكال مختلفة للاقراص المغنطة سوف نقوم بشرحها بشكل مفصل .

1– الأقراص الصلبة Hard Disks

تكون الأقراص الصلبة عادة على شكل حزمة من الأقراص Disk pack، تتكون كل جزمة من عدد من الأقراص المعدنية الرقيقة الدائرية الشكل وكلا وجهيها مغطى بطبقة من مادة سريعة المغنطة . وقطر القرص هو 14 بوصة، يحتوي الوجه الواحد على عدة مئات من المسارات (Tracks) الدائرة لها نفس المركز تبدأ من حافة القرص ، ولكل منها رقم خاص ، يمثل كل حرف أو رقم بمجموعة من الأرقام الثنائية مرتبة على نفس المسار ، وعلى الرغم من اختلاف أطوال المسارات

إلا أن كل منها يسمح بتخزين نفس العدد من الحروف، مما يعـــني أن المســـارات الخارجية تكون ذات درجة تركيز أقل من المسارات الداخلية والقريبة من مركـــز الدوران ، ينقسم كل مسار إلى عدد من القطاعات (Sectors) وترقـــم هـــذه القطاعات مثل المسارات . ويتم الوصول إلى مكان تخزين معـــين علـــى القـــرص بطريقة مباشرة وذلك عن طريق تحديد رقم القطاع ورقم المسار .

الأقراص الشائعة الاستخدام لها 100 / 200 / 400 مسار، يتم ترقيـــم هذه المسارات من الخارج إلى الداخل انظر الشكل (7) .

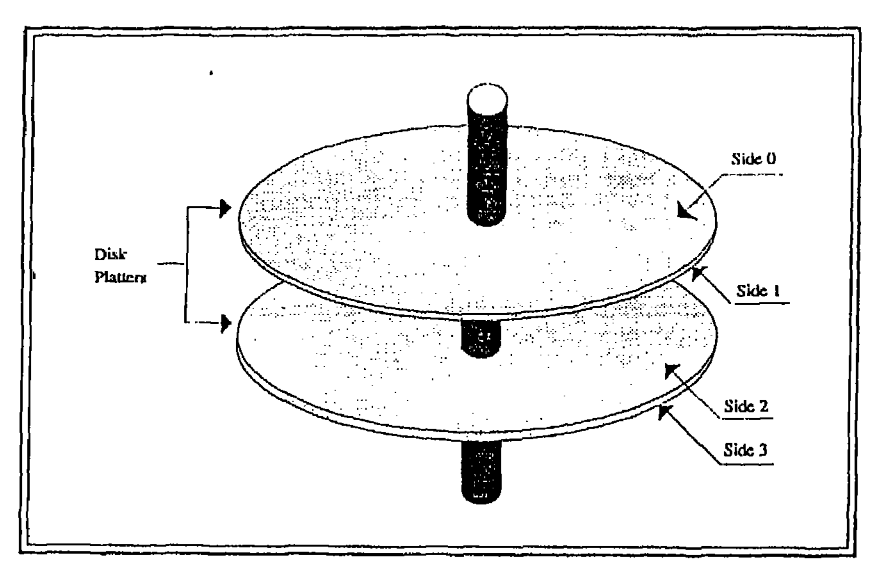


الشكل (7) يمثل قرص والمقاطع التي يتكون منها

يختلف عدد الأقراص في الحزمة من حاسوب إلى آخر ،كل قرص يسمى اســـطوانة (Platter)

ومن أنواع حزم الأقراص الصلبة الشـــائعة الاســـتخدام حزمـــة أقـــراص ذات5 اسطوانات وحزمة أقراص ذات6 اسطوانات وهي الأكثر شيوعاً ، وحزمة أقـــراص ذات 21 اسطوانة ، وتتميز حزم الأقراص الممغنطة بسمات تخزين عالية.

وحزم الأقراص إما أن تكون مثبتة بصفة دائمة ، أو أن تكون قابلة للاستبدال بأخرى وليست جميع الأقراص مثبتة في حزم . فهناك أجهزة أقراص تستخدم قرصاً واحداً قابلاً للاستبدال ويسمى قرص الخرطوشة Disk Cartridge وبعضها يستخدم اثنين أو اكثر ، تبلغ سعة التخزين للخرطوشة من 40 إلى 150 ميغابت، وبالتالي فإن إمكانية تغيير الأقراص تعني إمكانية زيادة التخزين .



الشكل (8) الأقراص المركبة

عادة تكون الأقراص مركبة على محور دوار وبين كل اسطوانتين مسافة 0.5 بوصة وتدور الأقراص بسرعة عالية ، تعتبر السرعات 200 لفة /دقيقة أو 2400 لفة / دقيقة من السرعات الشائعة ويتم التسجيل على كلا الوجهين لكل قرص كتجمعات من النقط الممغنطة ، انظر الشكل (8) السابق .

يطلق على وحدات تشغيل الأقراص بالسواقات (Disk Drivers) والتي أحد مكوناتها رؤوس القراءة والكتابة ، هناك نوعان من وحدات رؤوس القراءة والكتابة على الأقراص الممغنطة وهما الرأس المتحرك والرأس الثابت في حالة الرأس

المتحرك فإنه يتحرك أفقيا عبر سطح القرص بحيث يمكن الوصـــول إلى أي مســـار محدد، هناك رأس لكل سطح ، وتتحرك جميع الرؤوس معاً ، اما في حالة الـــرؤوس الثابتة فهناك رأس للقراءة والكتابة مخصص لكل مسار ، وبالتالي فليس هناك حاجـة لحركة الرأس كما أن الوصول إلى موقع تخزين محدد يتم بسرعة أكبر .

ليس هناك تلامس بين رؤوس القراءة والكتابة وأسطح الأقراص ، وإنما ترتكز هــذه الرؤوس على وسائد من الهواء ، حيث ان حركة الهواء الناتجة من دوران الأقـــراص تجعل رؤوس القراءة والكتابة قريبة للغاية من سطح القرص . لذلك قـــد تســـبب ذرات التراب في أتلاف بعض أجزاء من القرص ، كما تسبب في إفساد البيانـــات والمعلومات المخزنة عليها .

كيف يعمل القرص الصلب :

يمكن تلخص عمل القرص الصلب وكما يلي :

1. يطلب التطبيق البرمجي من نظام التشغيل .
2. تستقبل البرامج الخدمية لذاكرة القرص طلب القراءة والكتابة للتأكد مـــن أن البيانات المخزنة في الذاكرة الرئيسية مخصصة لذاكرة القرص الكـــاش . فـــاذا كانت البيانات المطلوبة موجودة في الذاكرة الكاش ، يتم نسخها إلى ذاكـــرة التطبيق الوسيطة Application Buffer ، وإذا لم تكن موجودة يحول طلب القراءة والكتابة إلى ضابط القرص الصلب لمعالجته .
3. يفحص ضابط القرص الصلب الذاكرة الوسيطة الموجودة على لوحة القـــرص لمعرفة فيما اذا كانت البيانات مخزنة هناك ، فإذا كانت كذلك تعاد البيانـــات إلى ذاكرة التطبيق الوسيطة .
4. اذا كانت البيانات غير مخزنة في ذاكرة القرص الصلب الوسيطة يقوم الضـــابط Controller بتشغيل المشغل الميكانيكي Actuator لرأس القرص الصلب.

5. يقوم مشغل رأس القرص بتحريك رؤوس القراءة والكتابة عبر الاطباق القرصية إلى المسار المعني .

6. يدور محور الدوران Spindle الموصول بسواقة محرك الاطباق القرصية ليساعد في جلب البيانات المطلوبة تحت رؤوس القراءة والكتابة .

7. تقوم رؤوس القراءة والكتابة بكتابة البيانات على الاطباق عن طريق تغيير القطبية .

8. يحتفظ ملف نظام التشغيل بسجل لمجموعة Clusters المستخدمة من قبل كل من ملف من الملفات .

2- أقراص ونشستر Winchester Disks

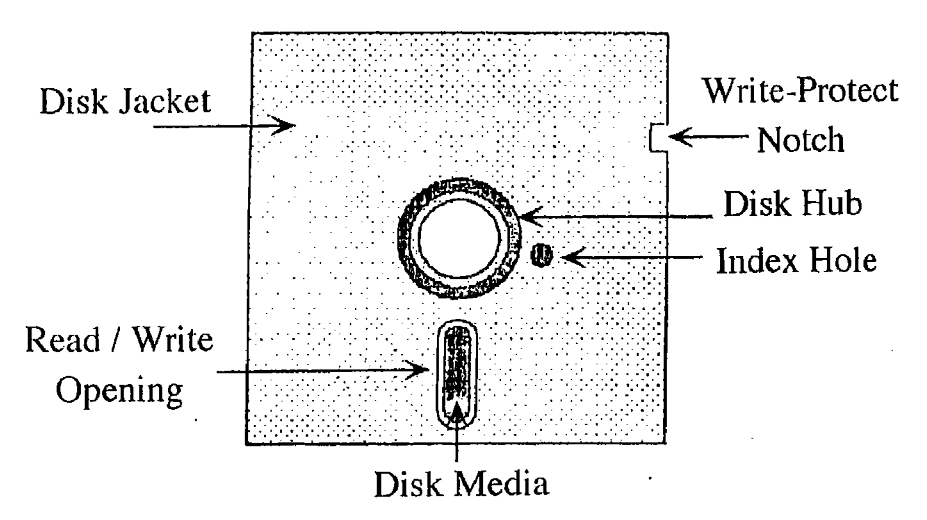
وهي عبارة عن أقراص محكمة العزل عن الهواء الخارجي ومصممة بحيث تستطيع رؤوس القراءة والكتابة الهبوط على سطح القرص . إذ يؤدي العزل عن الهواء الخارجي إلى منع التلوث بالغبار والدقائق الاخرى .

تعتبر أقراص ونشستر سريعة وقوية التحمل كما أنها منخفضة السعر مقارنة بالأقراص التقليدية الصلبة ، ونظراً لأنها محكمة الإغلاق فإنها لا تحتاج إلى صيانة دورية . الأحجام القياسية لهذه الأقراص هي2,25 بوصة و 14 بوصة وهي مختلفة السعات التخزينية .

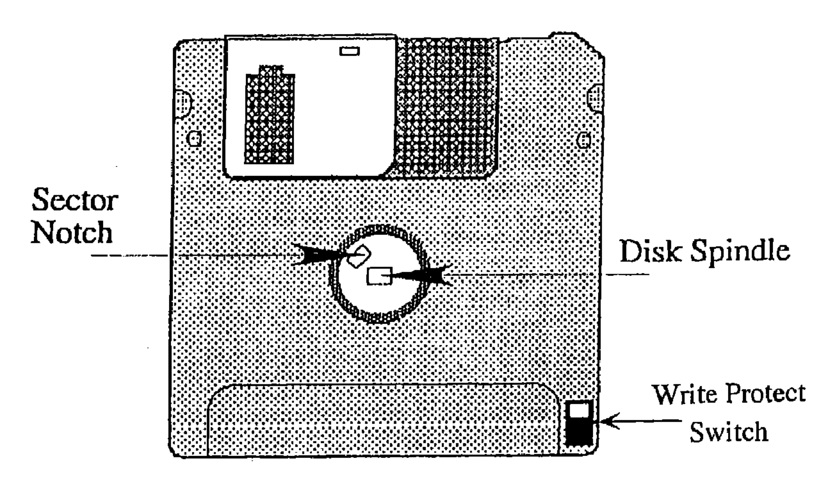
3- الأقراص المرنة Floppy Disks

يعتبر القرص المرن من وسائل التخزين الثانوية الشائعة الاستخدام وخاصة في الحواسيب الصغيرة ، والقرص مصنوع من نوع معين من البلاستيك وهو مغطى بمادة قابلة للمغنطة ومحفوظة داخل غلاف من البلاستيك وذلك لحماية السطح المغنط .

ينقسم كل قرص من الداخل إلى مجموعة من المسارات ، وكل مسار منـها ينقسم إلى عدد من القطاعات تتواجد هذه الأقراص بأحجام مختلفة 5.25 بوصـة و3.5 بوصة ، كما في الشكلين (9) و (10) على التوالي .



الشكل (9) يمثل القرص المرن 5.25 بوصة



الشكل (10) يمثل القرص المرن 3.5 بوصة

يتم التسجيل على وجهي القرص أو على وجه واحد ، ترقـــم المســـارات علـــى الأقراص المرنة القياسية من0 الى39 (40 مساراً) ، وترقم القطاعات مـــن 1 الى9 مما يعطي 360 قطاعاً إجمالياً (40 مسارا ضرب9 قطاعـــات في المســـار الواحد) على كل جانب ، ويتسع كل قطاع512 بايت لذلك تبلغ سعة قـــرص مرن منخفض الكثافة على كلا الجانبين (80*18*512*2=1,474,560) .

تمتلك الأقراص العالية الكثافة قياس 5.25 ،80 مساراً في كل منها 15 قطاعاً ، مما يعطي 1200 قطاعاً وبالتالي تكون سعته 1.228.800 بايت (1,2MB) .

تحتوي الأقراص المرنة قياس 3.5 بوصة على 80 مسارا في كل جـــانب و 18 قطاعا وبالتالي تكون سعتها 1,474,560 بايت (1.44 MB)بينمـــا تحتـــوي الأقراص المرنة ذات 36 قطاعا على 2,949,120 بايت (2,88 MB).

4- أقراص الليزر الضوئية Laser Optical Disk

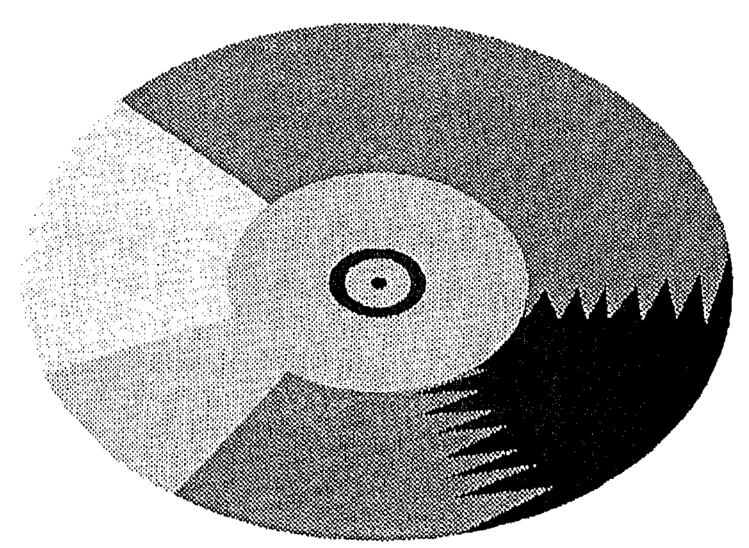
تعتبر أقراص الليزر الضوئية من أحدث أدوات التخزين الثانوية ، حيـــث تعتمد على تقنية تسجيل المعلومات من خلال أشعة الليزر الضوئية ، تستخدم هـــذه الأقراص لإدخال البرامج والمعطيات والبيانات الى الحاسوب ولكن التسجيل عليـــها يتم مرة واحدة وهذا ما ندعوه بالتسمية:

Read only memory – Copact Disk (CD)

تقوم هذه الأقراص بتخزين حجم هائل من البيانات يتراوح بين 440 MB وحتى 1.2 GB في الأقراص الحديثة .

القرص الليزري هو قرص دائري لا يزيد قطره على 4.51 أنش ، انظـــر الشكل (11) ، يتميز هذا القرص بأنه لا يمكن مسح البيانات المخزنة عليه ، كمـــا يتميز بمقاومة أفضل العوامل البيئية وهو لا يتأثر بالغبار وأكـــثر تحمـــلا للحـــرارة والبرودة من الأقراص المرنة .

يتم التوجه حاليا لاستخدام أقراص ضوئية قابلة للمسح والتسجيل ، وقد وضعت بعض الشركات تصميم لشكل آخر من الأقراص دعته DVD ولكن هذه الأقراص ما تزال مرتفعة الثمن وقليلة الاستخدام .



الشكل (11) يمثل القرص الليزري

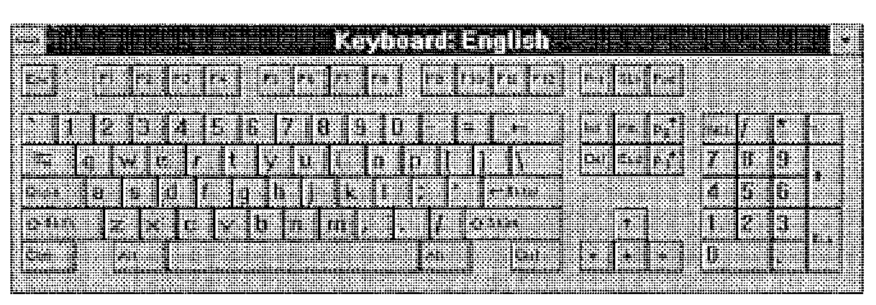
(2 – 5) وحدات الإدخال Input Units

تمثل وسائل الإدخال حلقة الربط بين مستخدمي الحاسوب ، والحاسوب نفسه ، حيث يستطيع مستخدم الحاسوب ادخال برامجه وبياناته إلى الحاسوب عن طريق وحدات ادخال متنوعة وتقوم وحدة الإدخال بتحويل الحروف والأرقام والعلامات الخاصة إلى مايناظرها من الاعداد الثنائية ثم ارسالها إلى الذاكرة الرئيسية. سوف نوضح أهم وسائل الإدخال الشائعة .

1- لوحة المفاتيح Keyboard

وهي تشبه لوحة المفاتيح الخاصة بالالة الكاتبة إلى حد كبير وتعتبر أشهر وحدات الإدخال المستعملة الان . وهناك أنواع كثيرة مختلفة من لوحات المفاتيح ولكن لوحة المفاتيح القياسية والأكثر انتشارا هي تلك التي يتم توزيع الحروف والأرقام والعلامات الخاصة بها طبقا لما هو متبع في الآلة الكاتبة القياسية والتي يطلق

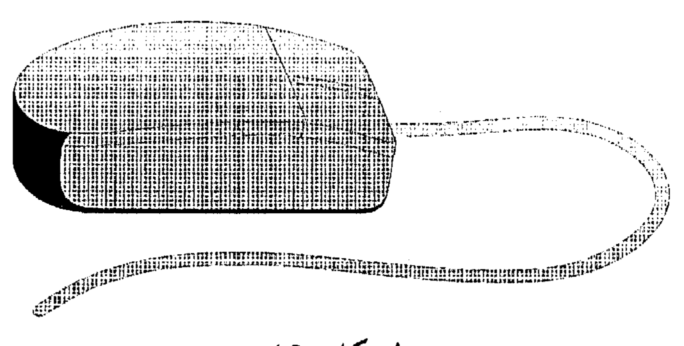
عليها لوحة المفاتيح كويرتي (Qwerty) والتي نشأ أسمها من مجموعـــه مفـــاتيح الحروف على الجانب الايسر من الصف الثالث من الاعلى . الشكل (12) تمثيـــل نموذجاً للوحة مفاتيح نموذجية لحاسب IBM .



الشكل (12) يمثل لوحة المفاتيح

2- الفأرة Mouse

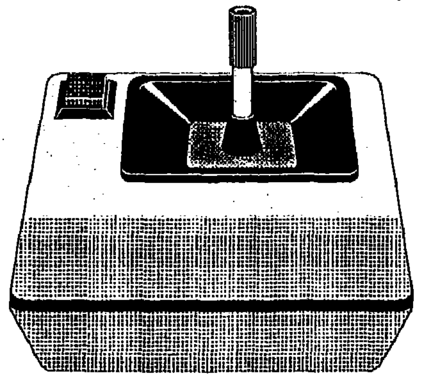
الفأرة عبارة عن جهاز تحكم الكتروني في حجم قبضة اليد يرسل إشـــارات إلكترونية إلى الحاسوب وذلك لتحريك مؤشر الشاشة(cursor) . ويتم ذلـــك بتحريك الفأرة على أي سطح الكتب أو المنضدة الموضوع عليها الشاشـــة ، ممـــا يؤديا إلى تحريك كرة دوارة موجودة أسفل الفأرة وبالتالي تحريك مؤشر الشاشة إلى الأمر الذي يكون مكتوباً أو مرسوماً على الشاشة وبالضغط على زر موجود علـــى سطح الفأرة العلوي يتم اختيار هذا الأمر ويقوم الحاسب بتنفيذه . يمكن الاستفادة من الفأرة أيضاً في عملية الرسم . الشكل (13) يمثل أحد نماذج الفأرة .



الشكل (13)

3_ عصا التحكم اليدوي (Joystick)

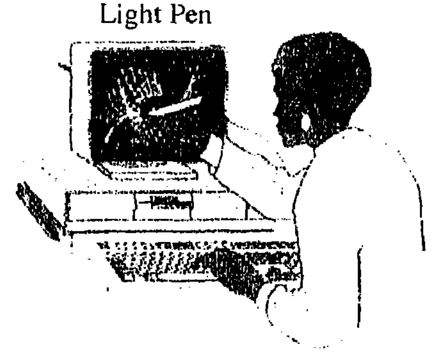
وتستخدم مع الألعاب بصفة خاصة لتوجيه أشياء معينه على الشاشة ، ويتم ذلك بواسطة تحريك ذراع صغير في الاتجاهات المختلفة . كما يوجد جهاز خاص يتم استخدامه في إطلاق القذائف ووسائل النيران المختلفة إذا كانت اللعبة تستدعي ذلك ، انظر الشكل(14) .



الشكل (14) يمثل نموذج لعصا التحكم اليدوي

4-القلم الضوئي (Light Pen)

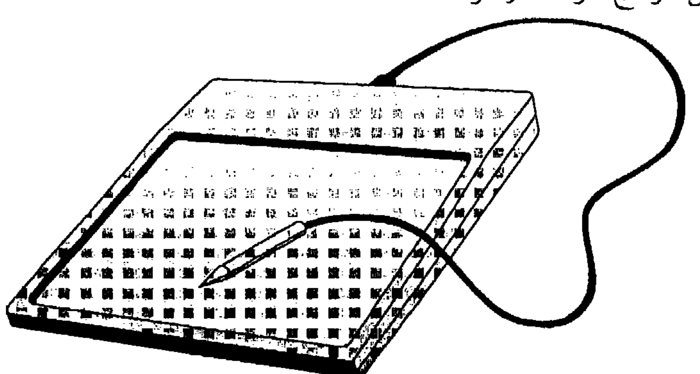
بملامسة هذا القلم لأي نقطة على الشاشة فان الحاسوب يقوم بقراءة موقع هذه النقطة ولذلك فهو يستخدم على نطاق كبير في التصميمات الهندسية وذلك لعمل تصميمات معقدة باستخدام الحاسوب وهو ما يطلق عليه تطبيقات تصميم بمساعدة الحاسوب (Computer Aided Design) واختصارا (CAD) .



الشكل (15) استخدام القلم الضوئي

5-لوحة الرسومات (Graphics Table)

وهي لوحة خاصة يمكن الرسم عليها وتنقل ما يرســـم مباشـــرة الى شاشـــة الحاسوب وتستخدم عادة في التصميمات الهندسية ومكـــاتب الرســـم . الشـــكل (16) يمثل نموذج للوحة الرسومات .



الشكل (16) يمثل لوحة الرسومات

6-أجهزة التعرف على الأصوات (Voice Recognition Devices)

استطاع بعض منتجي الحواسيب إنتاج أجهزة تمكن المستخدم من التحـــدث إلى الحاسوب . ويتم ذلك باستخدام ميكروفون ، الذي يقوم بتحويـــل الموجـــات الصوتية إلى موجات كهربائية يتم تحويلها إلى شفرات ثنائية ويتم مقارنتها بمعجـــم الألفاظ الصوتية ثم تسجيلة وتخزينة في ذاكرة الحاسوب . عندما يكون هناك توافـــق بين الكلمات الداخلة إلى الحاسوب مع تلك المخزنة في ذاكرته يقـــوم الحاســـوب باجراء العمليات المطلوبة ، معظم أجهزة التعرف على الأصوات والتي تم انتاجها في السبعينات كانت تعتمد على متكلم معين ولا يستطيع الحاسوب التعـــرف علـــى الكلمات الواردة من شخص آخر وذلك لاختلاف طبقات الصوت من شخص إلى آخر، في نهاية الثمانينات تم انتاج نظم عامة لا تعتمد طبقات صوت شخص معين، ولكن الكلمات التي يمكن التعرف عليها مازالت محدودة كما ان تكـــاليف هـــذه النظم ما زالت باهظة .

وقد قامت شركة IBM بتصميم نظام تعرف على الاصوات يمكن التعرف من خلاله على 6000 كلمة صوتية من تلك الكلمات الشائعة الاستعمال في المراسلات التجارية .

7- قارئ حروف الحبر المغنط

Magnetic Ink Character Reader (MICR)

يستخدم الحبر الممغنط في كتابة الحروف والأعداد والرموز الخاصة على شيكات البنوك والمصارف ، ويتم ذلك باستخدام اشكال قياسية للحروف والأرقام والأشكال الخاصة والتي يتم طباعتها بأحبار تحتوي على رقائق قابلة للمغنطة من أكسيد الحديد ، يوجد هناك نظامين لاشكال الكتابة بنظام الحبر المغنط، النظام الأول هو نظام (E/3B) ويتكون من الارقام من صفر إلى تسعة بالإضافة إلى أربعة من الرموز الخاصة وهو يستخدم في الولايات المتحدة الأمريكية وإنكلترا ، ويستخدم هذا النظام في كتابة شيكات البنوك ، حيث يتم طباعة الشفرة الخاصة للبنك ورقم الحساب للعميل بالحبر المغناطيسي ، وعند تقديم الشيك إلى البنك يتم كتابة قيمة الشيك في الجانب الأيمن السفلي من الشيك بنفس الحبر المغنط وبواسطة آلة خاصة ، وتدخل بيانات الشيك إلى الحاسوب عن طريق وحدة المعالجة المركزية (CPU) بالحاسوب مباشرة أو تخزينها على شريط ممغنط بواسطة آلة خاصة وتشغيلها فيما بعد ، كما يمكن لوحدة القراءة أن تقوم بفرز تلك الشيكات على أساس اسم البنك أو رقم الحساب أو أي أساس آخر ووضعها في جيوب خاصة بآلة القراءة ، يبلغ متوسط سرعة قراءة الشيكات بهذا الاسلوب حوالي 1200- 1500 شيك في الدقيقة الواحدة .

النظام الثاني للكتابة بنظام MICR هو نظام CMC7 ، وهو نظام نشأ أساساً في فرنسا ويستخدم في أوروبا ، وهو يحتوي على الأرقام من صفر إلى تسعة بالاضافة إلى الحروف الهجائية وخمسة رموز خاصة ، ويتم كتابة أي رمز منها باستخدام سبعة خطوط ممغنطة بينها ست مسافات متغيرة العرض ، المسافة الكبيرة تؤدي إلى

توليد الرقم الثنائي واحد ، بينما المسافة الصغيرة تؤدي إلى توليد الرقم الثنائي صفر.

تتميز الحروف المطبوعة بالحبر الممغنط بأنه يمكن قراءتها دائما حتى بعد طمسها بالاختام في حالة شيكات البنوك ، وقارئ MICR يستطيع فقط التأكد من الشيك لكن عملية التأكد من المبلغ المسحوب وشخصية القائم بالسحب والتأكد من صحة التوقيع مازالت أمور يتم القيام بها يدويا.

8- قارئ العلامات الضوئي

Optical Character Reader (OCR)

يوجد العديد من القارئات الضوئية منها ما هو ثابت على هيئة منضدة ، ومنها ما هو صغير الحجم يمكن الإمساك به باليد الواحدة وتمريره على كود الاعمدة الموجودة على السلعة وبذلك يتم التعرف على السلعة ومواصفاتها وسعرها ان معظم أجهزة قراءة الحروف الضوئية تكون مصممة لكي تقرأ الحروف المطبوعة والمكتوبة على الآلة الكاتبة والمكتوبة بخط اليد .

9- الشاشة الحساسة للمس Touch Sensitive Screen

عن طريق لمس الشاشة يمكن توجيه الحاسوب لتنفيذ أوامر معينة .

(6-2) وحدات الإخراج Output Units

أجهزة الإخراج هي الأجهزة والوسائط التي تقوم باستقبال نتائج معالجة الحاسوب للبيانات الداخلة وتجهيزها بالصورة المطلوبة وعرضها على المستخدم ، سوف نلقي الضوء على أجهزة الإخراج الموجودة حالياً .

1- الشاشة Screen

ويطلق عليها أيضاً وحدات العرض المرئي Visual Display Units كما يطلق عليها اسم Visual Display Terminal ، وتعتبر من أشهر وسائل الإخراج المستخدمة ، كما تعتبر شاشة العرض من الأجزاء الرئيسية والهامة

للحاسوب إذ بواسطتها يمكن عرض وملاحظة النتائج التي يعالجها الحاسوب ويمكن أن يظهر على الشاشة نتائج رقمية أو نتائج جدولية أو رسوم وغيرها من الاشكال.

يمكن أن يجهز الحاسوب بشاشة عادية أو شاشة رسم ، وتكون الشاشة العادية ذات ألوان أبيض وأسود ، يستخدم هذا النوع من الشاشات لعرض الأرقام والأحرف والرموز ،ويمكن أن نشكل بواسطة بعض الرموز الموجودة على لوحة المفاتيح جداول وأشكال حسب الإمكانية ، يعتبر هذا النوع من الشاشات أقل ضرراً على الرؤية من النوع الثاني (شاشات الرسم) لذلك يفضل في كثير من الاحيان هذا النوع من الشاشات على شاشات الرسم ، أخيراً يطلق على هذا النوع من الشاشات اسم شاشات احادية اللون (Monochrome).

أما النوع الثاني فهو شاشات الرسم ويمكن أن تستخدم في كلا النظامين سواء كان رسم أو عرض نتائج كما في الشاشة العادية ، وهي في أغلب الأحيان ذات ألوان مختلفة يمكن التحكم بالألوان بواسطة أوامر خاصة بكل نظام من نظم البرمجة .

يكون حجم الشاشة أو قياسها في نظام عرض النتائج 25سطر و 80 عمود وبذلك يمكن إظهار حوالي 2000 حرفاً على الشاشة ، أما في نظام الرسم فتكون ذات قياسات600 عمود ، و200 .

والجدول التالي يبين أنواع الشاشات الأكثر انتشاراً :

نوع الشاشة	نظام عادي	ألوان	نظام الرسم	ألوان
MDA (Monochrome Display Adapter)	80x25	2	640x200	2
CGA (Color/graphics Adopter)	80x25	16	640x200	16
Hercules	80x25	2	720x340	2
EGA(Enhanced Graphics Adapter)	80x25	16	640x200	16
MCGA(Multicolor Graphics Array)	80x25	256	640x200	256
VGA (Video Graphics Array)	80x25 80x50	256	640x480	256
SVGA(Super Video Graphics Array	80x25 80x50	256	640x480	256

من أهم الصفات المميزة للشاشة سرعة عرض المعلومات في النظام العـــادي، كل الشاشات تقريبا تعمل بسرعة عالية لكن في نظام الرسم يختلف الأمـــــر مـــن شاشة الى أخري حسب نوعها .

2- الطابعات Printers

هناك المئات من الطابعات المختلفة المتوفرة لعائلة الحواســـيب الشـــخصية تؤدي جميعها نفس المهمة ولكن هناك أنواع مختلفة من الطابعات تعطيك درجـــات مختلفة من النوعية والأداء . والفئتان الرئيستان لطابعات الحواسيب الشخصية هـــي الطابعات النقطية (Dot Matrix) والطابعات اليزرية (Laser Printers) .

أ. الطابعات النقطية Dot Matrix

تعمل الطابعات النقطية بإطلاق سلسلة من الدبابيس المعدنية المرتبة في صـــف أفقي، تضغط هذه الدبابيس على شريط الحبر الذي ينشيء بدوره نقـــط علـــى الورق . وخلال تحريك رأس الطباعة (الأداة التي تحتوي على الدبابيس) ذهابـــاً وإياباً على الصفحة تقوم الدبابيس بانشاء نمط معين هو عبارة عن صفوف مـــن النقاط والتي تتألف منها الحروف .

حوالي عام 1985 زادت شركات التصنيع عدد الدبابيس في رأس الطباعة من 9 الى 24 . حيث وضعوا صفين من الدبابيس الصغيرة (12 دبوس في الصـــف الواحد) عوضاً عن صف واحد من الدبابيس الأكثر ثخانة ، وقد كانت النتيجـــة فعالة ومثيرة للإعجاب .

تستطيع الطابعة النقطية المؤلفة من 24 دبوس ان تعطي نوعية جيدة مـــن الخط (Near Letter Quality) و اختصاراً (NLQ) وهي الحروف التي تقارب حروف الآلة الكاتبة من حيث الجودة .

تطبع الطابعة النقطية 80 حرف في السطر الواحد و 6 اسطر في البوصة ، كمـــا تستطيع طباعة 132 حرف في السطر و 18 سطر في كل بوصة .

تستطيع الطابعة النقطية الجيدة ان تطبع 80 حرفا في الثانية الواحدة ، اما الطابعات الاسرع فبإمكانها طباعة حتى 160 حرفا في الثانية او حتى 240 حرفا في الثانية .

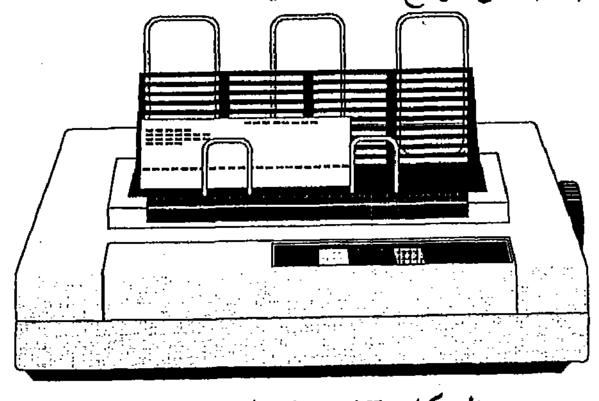
تجهز الطابعات النقطية بإحدى آليات تغذية الورق التالية :

- تغذية باسطوانة او التغذية الاحتكاكية .
- تغذية بدبابيس .
- تغذية بجرار .

التغذية بالاسطوانة تقدم الورق مثلما يتم في الآلة الكاتبة ، وتقوم بوضع الورق في الطابعة وإدارة المقبض الى الأمام وضبط وضعية الورق تحت رأس الطباعة، وهذه الطريقة هي المثلى لطباعة صفحة واحدة او لطباعة المغلفات .

تستعمل آليات التغذية بدبابيس أو بجرار الصفحات المنطوية المتصلة في الطابعة ، وهذا الورق من النوع المزود بثقوب على الجانبين والتي تتطابق مع دبابيس الطابعة المستعملة لتحريك الورق في الطابعة .

أخيراً لا تزال الطابعات النقطية أكثر إنتشاراً بين طابعات الحواسيب وتختلف اسعارها حسب نوعية وسرعة الطباعة وعرض حاملة راس الطباعة ، وكذلك حسب امكانية التحكم بألوان الطباعة وبنوع الخط ، كما تتعلق أسعارها بحجم ذاكرتها RAM التي تستعمل كمخزن وسيط Buffer لحفظ الأحرف الداخلة الشكل (17) يمثل نموذج لطابعة نقطية .



الشكل (17) يمثل طابعة نقطية

ب. الطابعات الليزرية Laser Printers

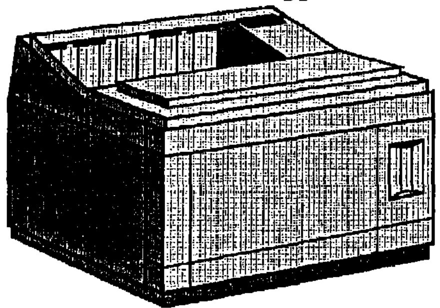
الطابعات الليزرية هي أكثر الطابعات قوة وأكثرها كلفة ، ولكن منذ عـام 1989 تم تخفيض اسعار البعض منها .

تنشأ الصورة في الطابعة الليزرية بواسطة شعاع ليزر ، حيث تقوم دوائـــر إلكترونية معقدة بالتحكم بهذا الشعاع ، يرسم الشعاع صورة على الاســطوانة داخل الطابعة الليزرية بحيث تنشأ شحنة كهربائية حيثما يلامس الشعاع يلامس الشعاع الاسطوانة تدور الاسطوانة تحت خرطوشة حبر التصوير ملتقطة الحـــبر على اجزاء الاسطوانة المشحونة كهربائيا ، وعندما تلامس الاسطوانة الورق يتم نقل الصورة الى الورق ، وتستعمل الحرارة بعد ذلك لصـــهر وتثبيـــت حـــبر التصوير على الورق .

تمتلك الطابعة الليزرية ذاكرة RAM ، حيث تتعلق بها ســـرعة الطباعـــة فكلما زاد حجم الذاكرة كلما زادت سرعة الطباعة ، كما تقوم بحفظ الـــبرامج التي يجب ان تطبع .

تتفوق الطابعة الليزرية على الطابعة النقطية أو أي نوع آخـــر مـــن طابعـــات الحاسوب الشخصي وهي الطابعة المثالية التي تعطي نصا بجودة عالية .

لقد كانت الطابعات الليزرية الأولى تطبع أربع صفحـــات في الدقيقـــة ولكن الطرازات الحديثة تصل الى أكثر من 12 صفحة في الدقيقة الشكل (18) يمثل نموذج لاحدى الطابعات الليزرية .



الشكل (18) يمثل الطابعة الليزرية

3- الراسم Plotter

يقوم بإخراج النتائج في هيئة رسوم أو صور بيانية ، وهناك وحدات رسم تستخدم الأقلام (Pen Plotters) ومنها ما يستخدم اسطوانة (DRUM) وقاعدة مستوية (Flat Pen) ، وهناك وحدات رسم اخرى تستخدم راسمات الحبر النفاث (Inkjet Plotters) وهي قادرة على عمل رسومات ذات حجم كبير وتستخدم الاسطوانة كحامل للورق .

يمكن للرسام ان يقوم برسم النتائج بالألوان وذلك باستخدام مجموعة من الأقلام الملونة او راسمات الحبر النفاث ا لملونة . والراسم عادة بطئ ولكنه يتميز بدرجة عالية من الدقة وهي خاصية اهم من السرعة خاصة في التطبيقات الهندسية التي تتطلب دقة عالية .

4- الذراع الروبوتي Robotic Arm

تستخدم اذرع الروبوت حاليا بكثرة في معظم خطوط إنتاج المصانع الكبرى. يتم التحكم في الحركات الاساسية للذراع الروبوتي بواسطة الحاسوب، وتختلف طريقة وشكل الذراع طبقا لطبيعة وطريقة استخدامها .

بالإضافة إلى الأذرع الروبوتية فهناك العديد من الماكينات مثل ماكينات الخراطة والفريزة والتثقيب وماكينات الغزل والنسيج وخلافه يتم التحكم فيها بواسطة الحاسوب وهي تعتبر وحدات إخراج لنظام الحاسوب المستخدمة به .

5- الميكروفيلم Micro Film

يستخدم لتسجيل معلومات على شرائط افلام قياس 16 او 35 مليمتر بعد تصغير كل صفحة في الحجم بما يزيد عن 48 مرة.

6- التعامل مع الاصوات

ويتم ذلك باستخدام نظام الاستجابة للاصوات حيث يمكـــن اســتخدام الحاسوب لإجراء حوار او اتصال ، وتعتبر هذه الطريقـــة مناســبة إذا كــان المطلوب مجرد معرفة إجابات تقليدية عن استفسارات حول معلومات معينــــة، ويتم ذلك عن طريقة إدخال الاستفسارات أو الرسائل وإرسالها عبر خطـــوط تليفونية الى حاسوب مركزي ، حيث يقوم نظام الإستجابة الصوتية بــــتركيب الاجابة من العبارات و الكلمات المسجلة و المخزنة مسبقاً ، ثم ترسل الاجابـــة الصوتية على الاستفسارات .

تمارين وأسئلة عامة

1- اذكر الوحدات الأساسية التي يتكون منها الحاسوب ؟

2- اذكر مكونات وحدة التشغيل المركزية ومكونات ووظيفة كل وحدة ؟

3- اشرح باختصار دورة المعالجة في وحدة التحكم ؟

4- اذكر أهم وحدات التخزين الداخلية ؟

5- ما هي مناطق الذاكرة الرئيسية ؟

6- ما هي اهم معايير التمييز بين الذاكرات ؟

7- تكلم عن الذاكر الفورية ؟

8- ما هي اشكال الذاكرة ROM ؟

9- عدد أنواع المسجلات ووظيفة كل منها ؟

10- تكلم عن المخازن الانتقالية ؟

11- اشرح آلية عمل القرص الصلب ؟

12- عدد أهم أنواع وحدات الإدخال ؟

13- عدد أهم أنواع وحدات الإخراج ؟

14- تكلم بإختصار عن القرص المرن 3.5 ؟

الفصل الثالث

الأنظمة العددية

Numbering Systems

(3 – 1) ماهية النظم العددية :

(3 – 1 – 1) مفاهيم النظم العددية :

استخدمت البشرية الأعداد منذ قديم الزمان ،فقد كان الراعي يحتاج الى نظام بسيط ليعد أغنامه فاستخدم أصابع يده العشرة أو ربما الحصى المتوفرة حوله، واستخدم المصريون قبل الميلاد بـ 3400 سنة نظام العد العشري الذي نستخدمه نحن الآن ، ويعتقد أن سبب بناء نظام العد هذا على العدد عشرة ، يرجع الى أن بداية تعداد الأشياء كان باستخدام أصابع اليد العشرة .

إن النظام الذي اعتدنا الحساب به ليس الوحيد المستخدم في الوقت الحاضر فتوجد قبائل السكان الاصليين في استراليا وقبائل في بعض أجزاء غينيا الجديدة تستخدم النظام الثنائي ، أي نظام العد الذي فيه رقمين اثنين فقط ، إما صفراً أو واحد كما توجد قبائل في أمريكا الوسطى والمكسيك تستخدم نظام عد خماسي. وهذا ناتج عن احتياجهم لنظام عددي بسيط يناسب حضارتهم البدائية . يوجد أيضاً بعض القبائل في أمريكا الجنوبية والمكسيك تستعمل نظام عددي ذو أساس يتكون من عشرين رمزاً نتيجة لاهتمامهم بالفلك والنجوم .

يوجد هناك خمسة عشر نظاماً للعد الجدول (1) ، يبين بعضـــها لا يـــزال مستخدماً لحد الآن بينما البعض الآخر لا يتعدى كونه نظاماً مدرجـــاً في كتـــب الباحثين والمؤلفين . سوف نتعرض في هـــذا الفصـــل للأنظمـــة المســـتخدمة في الحواسيب.

أعداد النظام و رموزه	النظام بالعربية	النظام بالإنجليزية	الأساس
01	الثنائي	Binary	2
012	الثلاثي	Ternary	3
0123	الرباعي	Quaternary	4
01234	الخماسي	Quinary	5
012345	السداسي	Senary	6
0123456	السباعي	Septenary	7
01234567	الثماني	Octenary	8
0123456678	التساعي	Nonary	9
0123456789	العشري	Denary	10
0123456789A	الاحد عشري	Undenary	11
0123456789AB	الأثنين عشري	Duodenary	12
0123456789ABC	الثالث عشري	Trenary	13
0123456789ABCD	الرابع عشري	Quatuordenry	14
0123456789ABCDE	الخامس عشري	Quinadenary	15
0123456789ABCDEF	السادس عشري	Hexadenary	16

جدول (1) الأنظمة العددية الخمسة عشر

(3 – 1– 2) النظام العشري Dicimal System

يحتوي النظام العشري كما رأينا في الجدول أعلاه على عشرة أرقام أو رموز تمثل الأعداد الصحيحة من الصفر حتى التسعة، وعليه فإن اساس النظام العشري هو b=10 ، من المعروف أن أي عدد من النظام العشري يتكون من بعض أو كل القيم الموقعية التالية : آحاد ، عشرات ، مئات ، ...

ولإيجاد قيمة أي عدد نضرب كل رقم بقيمته الموقعية ، ثم نجمع نواتج الرموز المختلفة .

فمثلا : العدد 6105 يتألف من :

الرقم 5 آحاد

الرقم 0 عشرات

الرقم 1 مئات

الرقم 6 آلاف

إذاً يمكن كتابته بالشكل التالي :

$$6105 = 6*1000 + 1*100 + 0*10 + 5*1$$
$$= 6*10^3+1*10^2+0*10^1+5*10^0$$

يدعى الشكل الأخير بالشكل الموسع (Expansion Form) ، حيث ضربنا كل رقم بقيمته الموقعية ، والقيمة الموقعية هي الأساس 10 مرفوعا الى الأس المساوِ الى موقع العدد :

الآحاد موقعه صفر ، العشرات موقعه 1 ، المئات موقعه 2 ، ...

وبناءاً على ما سبق فإنه في أي نظام عددي يمكن كتابة العدد بالشكل الموسع كما يلي :

نضرب كل رقم بقيمته الموقعية ثم نجمع نواتج الرموز المختلفة ، والقيمة الموقعية للرقم هو أساس النظام مرفوعاً للأس المساوِ لموقع الرقم في العدد .

فإذا كان العدد مؤلف من n رقم على يسار الفاصلة (القسم الصحيح) وm رقـــم على يمين الفاصلة (القسم الكسري) عندئذ :

0 : يعبر عن موقع الرقم الأول الذي يأتي على يسار الفاصلة .

1 : يعبر عن موقع الرقم الثاني الذي يأتي على يسار الفاصلة .

2 : يعبر عن موقع الرقم الثالث الذي يأتي على يسار الفاصلة .

.....................................:

.....................................:

n-1 : يعبر عن موقع الرقم الـــ n الذي يأتي على يسار الفاصلة .

-1 : يعبر عن موقع الرقم الأول الذي يأتي على يمين الفاصلة .

-2 : يعبر عن موقع الرقم الثاني الذي يأتي على يمين الفاصلة .

.....................................:

.....................................:

-m : يعبر عن موقع الرقم الـــ m الذي يأتي على يمين الفاصلة .

والصيغة العامة للشكل الموسع للعدد x في نظام عددي أساسه b هي :

$$X = (\dots d_3\ d_2\ d_1\ d_0\ ,\ d_{-1}\ d_{-2}\ d_{-3} \dots)_b =$$
$$= \dots + d_3 b^3 + d_2 b^2 + d_1 b^1 + d_0 b^0 + d_{-1} b^{-1} + d_{-2} b^{-2} + d_{-3} b^{-3} + \dots$$

حيث d يعبر عن الرقم في الموقع ، b أساس النظام .

مثال (1) : العدد 837.526 يعبر عنه بالشكل الموسع التالي :

$$837.524 = 8*10^2+3*10^1+7*10^0+5*10^{-1}+2*10^{-2}+6*10^{-3}$$
$$= 800+30 + 7 + 5/10 + 2/100 + 6/1000$$

(3-1-3) النظام الثنائي والعمليات الحسابية عليه Binary System

(1-3 -1-3) مقدمــة

إن العديد من العناصر الالكترونية للحاسوب ما هي الا عنـــاصر ثنائيـــة الاستقرار Bistable في طبيعتها ، بمعنى أنها تكون في احـــدى الحـــالتين فقـــط (كحالة سريان التيار فيها ON أو انقطاع التيار فيها OFF) ، إن هاتين الحـــالتين الممكنتين يشار اليهما بـــ (0) و(1) والتي هي الرموز المستخدمة في النظام الثنائي، عدا عن ذلك فإن أي وحدة مستقلة من المعلومات تمثل عادة في الحاسوب بواســطة سلسلة من هذه الرموز الثنائية Binary Digits التي تدعى اختصاراً بالانجليزيـــة BITS وهي اختصاراً للكلمتين حد ثنائي .

ان مثل هذه السلسلة يمكن النظر اليها على انها أعداد ثنائية وكثير من الحواســـيب تستخدم نظام الأعداد الثنائية .

إذاً النظام الثنائي هو نظام يعتمد على الأساس b= 2 إن الأعداد الثنائيـــة التي لا تحتوي على قسم كسري تدعى بالأعداد الثنائية الصحيحـــة Binary Integers وقوى الأساس في النظام الثنائي للقسم الصحيح هي :

$$2^0, 2^1, 2^2, 2^3,....$$

وأن قوى الأساس للقسم الكسري هي :

$$2^{-1}, 2^{-2}, 2^{-3},....$$

(3-1-3-2) التحويل من النظام الثنائي الى النظام العشري

Binary-Decimal Interconversion

لتحويل العدد في النظام الثنائي الى النظام العشري نتبع الخطوات التالية :

1. نكتب العدد بالشكل الموسع .
2. نوجد قيمة كل حد في الشكل الموسع
3. نوجد حاصل جمع الحدود .

مثال (2) :

$$(110101)_2=1*2^5+1*2^4+0*2^3+1*2^2+0*2^1+1*2^0$$
$$=32+16+0+4+0+1$$
$$=(53)_{10}$$
$$(101.1101)_2=1*2^2+0*2^1+1*2^0+1*2^{-1}+1*2^{-2}+0*2^{-3}+1*2^{-4}$$
$$=4+0+1+0.5+0.25+0+0.0625$$
$$=(5.8125)_{10}$$

والجدول(2) يحتوي على التمثيل الثنائي لبعض القيم الصحيحة و الجدول (3) يحتوي على قيمة الأساس المرفوع الى أس .

العدد الثنائي	العدد العشري
Binary Number	Decimal Number
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16
100000	32
1000000	64
10000000	128

الجدول(2)

القيمة العشرية	الأساس2 مرفوع الى أس
Decimal Number	Power of Two
1024	2^{10}
512	2^9
256	2^8
128	2^7
64	2^6
32	2^5
16	2^4
8	2^3
4	2^2
2	2^1
1	2^0
½ = 0.5	2^{-1}
¼ = 0.25	2^{-2}
1/8 = 0.125	2^{-3}

الجدول(3)

3 – 1 – 3 – 3) التحويل من العشري الى الثنائي
Decimal-Binary Interconversion

يمكننا ايجاد التمثيل الثنائي لرقم عشري N بتحويل جزئه الصحيح N_I وجزئه الكسري N_F كل جزء بصورة مستقلة . سنوضح العملية بالمثال التالي :

مثال (3) : حول العدد العشري $N = 109.78125$ الى مكافئة الثنائي .

أ- نحول القسم الصحيح $N_I = 109$ الى مكافئة الثنائي ، من أجل ذلك نقسم N_I على الأساس 2 ونكتب القسم الصحيح من ناتج القسمة في عامود نـاتج القسمة ، والباقي نضعه في عامود البواقي ثم نقسم حاصل القسمة علـى 2 ... وهكذا مع تسجيل الباقي بعد كل عملية تقسيم على اليمين .

التقسيمات Divisions	ناتج القسمة Quotients	البواقي Remainders
109÷2	54	1 ↑
54÷2	27	0
27÷2	13	1
13÷2	6	1
6÷2	3	0
3÷2	1	1
1÷2	0	1

نلاحظ ان بواقي القسمة لا يمكن إلا أن تكون مساوية للواحد أو الصفر ما دامـت عملية التقسيم تتم على 2 .

إن سلسلة البواقي مقروءة من الأسفل الى الأعلى (كما يشـير السـهم) تعطـي المكافئ الثنائي المطلوب ، و بالتالي :

$$N_I = (109)_{10} = (1101101)_2$$

ب- أما لتحويل القسم $N_F = 0.78125$ الى مكافئة الثنائي فيتم بضرب N_F بالأساس 2 ونأخذ من ناتج الضرب القسم الصحيح الذي سيكون إما صفـــراً أو واحد ونضعه في عامود الأقسام الصحيحة ، ثم نأخذ الجزء الكسري مـــن ناتج الضرب ونضربه بـــ 2 ، وهكذا نعيد ما سبق .

عمليات الضرب Multiplications	الاقسام الصحيحة Integral Parts
0.78125x2 = 1.56250	1
0.5625 x 2 = 1.1250	1
0.125 x 2 = 0.250	0
0.250 x 2 = 0.50	0
0.50 x 2 = 1.00	1

إن القسم الكسري الصفري في ناتج الضرب بـــــ 2 يشـــير الى نهايـــة الحسابات . نلاحظ أن القسم الصحيح لأي جداء لا يمكن أن يكون إلا صفـــراً أو واحداً ما دمنا نضاعف رقماً أقل من الواحد إن سلسلة حدود القسم الكسري مـــن الأعلى إلى الأسفل كما هو مشار إليها بالسهم تعطي المكافئ الثنائي المطلوب . أي أن :

$$N_F = (0.78125)_{10} = (0.11001)_2$$

وعليه فإن المكافئ الثنائي الكلي للرقم العشري N هو مجموع المكافئين أي :

$$N = N_I + N_F$$

$$(109.78125)_{10} = (1101101.11001)_2$$

(3- 1- 3-4) العمليات على النظام الثنائي

أولاً : جمع الأعداد الثنائية Binary Addition

تجمع الأعداد الثنائية وفق القواعد المبينة في الجدول التالي :

0+0 = 0		
0+1 = 1		
1+0 = 1		
1+1 = 0 ;	Carry 1	مع نقل باليد واحد
1+1+1 = 1 ;	Carry 1	مع نقل باليد واحد

مثال (4) : احسب المجموع الثنائي للعددين الثنائيين التاليين :

110011101 + 10110111

الحل : لنكتب العددين على الشكل التالي :

```
  110011101
+
   10110111
 ___________
 1001010100
```

مثال (5) : احسب المجموع الثنائي للأعداد الثنائية الاربعة التالية :

1001+ 1101 + 110 + 1011

الحل : نجمع العددين الأول و الثاني ثم نجمع الى المجموع الناتج العـــدد الثـــالث ثم نجمع الى المجموع الناتج العدد الرابع فتحصل على القيمة الاجمالية أي :

```
  1001
+ 1101
 10110
+  110
 11100
+ 1011
100111
```

مثال (6) : احسب المجموع الثنائي التالي :

$$11011.01 + 101.1101$$

الحل :نكتب الأعداد الثنائية بحيث تقع الفواصل الثنائية التي تفصل القسم الكسري عن القسم الصحيح تحت بعضها ثم نجمع :

```
   11011.01
+    101.1101
 100001.0001
```

ثانياً : ضرب الأعداد الثنائية Binary multiplication

يتم ضرب الأعداد الثنائية وفق القواعد المبينة في الجدول التالي :

0 * 0 = 0
0 * 1 = 0
1 * 0 = 0
1 * 1 = 1

مثال (7) : أوجد جداء العددين التاليين :

```
      1101011
    *   10110
      0000000
     1101011
    1101011
   0000000
 +1101011
 100100110010
```

مثال (8) : احسب الجداء الثنائي التالي $11.01*101.1$:

الحل :

```
     11.01
   * 101.1
   -------
      1101
     1101
    0000
   1101
  10001.111
```

نلاحظ وجود ثلاثة أرقام بعد الفاصلة وبالتالي هي نفس الطريقة المتبعة في النظـــام العشري .

ثالثاً : الطرح الثنائي Binary Subtraction

يتم طرح الأعداد الثنائية وفق القواعد المبينة في الجدول التالي :

$0 - 0 = 0$
$1 - 0 = 1$
$1 - 1 = 0$
$0 - 1 = 1$; (Borrow 1) مع الاستعارة واحد من العمود التالي

القاعدة الاخيرة تأتي من

$1+1 = 10$ وكذلك $10 - 1 = 1$

وبالتالي الفرق $1 - 0$ يتطلب الاستعارة التي تعطي $10 - 1 = 1$

مثال (9) : أوجد حاصل طرح الأعداد التالية :

$$11101 - 1011 =$$

الحل : نكتب الأعداد بحيث تكون الأعداد مرتبة في كلا العددين تحـــت بعضـــها البعض ابتدأً من جهة اليمين ، أي على الشكل التالي :

$$
\begin{array}{r}
11101 \\
-\ \underline{1011} \\
10010
\end{array}
$$

مثال (10) : أوجد حاصل طرح الأعداد التالية :

$$1100101001 - 110110110 =$$

الحل :

$$
\begin{array}{r}
1100101001 \\
\underline{-\ 110110110} \\
101110011
\end{array}
$$

نلاحظ في العمود الثاني وفي العدد الأول وجود صفر وفي العمود الثالث أيضا وجود صفر لذلك سوف نستعير الواحد من العمود الرابع وبالتالي تصبح قيمة العمود الثالث الذي اخترناه اثناء الاستعارة مساوياً الواحد وتصبح قيمة العمود الرابع صفر والعمود الثاني 10 .

مثال (11) : أوجد حاصل طرح الأعداد الثنائية التالية :

$$1101.101 - 11.10111 =$$

الحل : يجب أن نضع الأعداد الثنائية في كلا العددين بحيث تقع الفواصل تحت بعضها البعض ، أي :

$$
\begin{array}{r}
1101.10100 \\
\underline{\quad 11.10111} \\
1001.11101
\end{array}
$$

لا حظ أننا أضفنا في العدد الأول صفرين الى الجزء الكسري على يمين الفاصلة وذلك من أجل أن يصبح عدد الأعداد على يمين الفاصلة في العدد الأول مساوياً لعدد الأعداد على يمين الفاصلة في العدد الثاني .

رابعاً : التقسيم الثنائي **Binary Division**

القواعد المتبعة في تقسيم الأعداد الثنائية هي نفسها المتبعة في تقسيم الأعداد العشرية سواء أكانت صحيحة أم كسرية .

مثال (12) : أوجد ناتج ما يلي :

$$101001 \div 11 =$$

الحل :

```
                11011   <---- ناتج القسمة
              ________
المقسوم عليه ---> 11 | 1010001    <---- المقسوم
               - 11
              ------
                100
               - 11
              ------
                 0100
               -   11
              ------
                  011
               -   11
              ------
                   00   <---- باقي القسمة
```

وبالتالي حاصل القسمة هو العدد : 11011 : أي أن :

$$1110111 \div 11 = 11011$$

مثال (13) : أوجد ناتج ما يلي :

$$1110111 \div 1001 =$$

الحل :

```
              1101
1001 |  1110111
       - 1001
         1011
       - 1001
           10
```

اذاً حاصل القسمة هو 101 والباقي 10 أي أن :

$$1110111 \div 1001 = 1101$$

و الباقي 10 .

مثال (14) : أوجد ناتج ما يلي :

$$111.00001 \div 1.01 =$$

الحل : ننقل الفاصلة الثنائية بين القسم الصحيح والكسري في كل من القاسم والمقسوم عليه بمقدار منزلتين لتحويل القاسم 1.01 الى عدد صحيح ثم نجري عملية القسمة :

```
        101.101
101 |  11100.0001
      - 101
        1000
      -  101
          101
      -   101
          000
```

وبالتالي نجد ان حاصل القسمة 101.101 أي :

$$111.00001 \div 1.01 = 101.101$$

(3 – 1 – 4) نظام العد الثماني Otcal System

(3 – 1 – 4 – 1) مقدمــة

إن نظام العد الثماني له أساس b= 8 وأعداده الثمانية هي :

0 , 1 , 2 , 3 , 4 , 5 , 6 , 7

وهذا يختلف عن نظام العد العشري باختفاء الرموز 8 و 9 وبذلك يـــأتي بعد العدد7 العدد 10 وبعده يأتي العدد11 وهكذا حتى العدد 17 حيث يليه العـدد 20 ومن ثم 21 حتى 27 وهكذا ... وبالتالي نجد ان التعداد باستعمال رموز النظام العد الثماني هو كالآتي :

0,1,2,3,4,5,6,7
10,11,12,13,14,15,16,17
20,21, 22,23,24,25,26,27
30,31, ...

(3–1–4–2) التحويل من النظام الثماني الى النظام العشري

Octal-Decimal Interconversion

لقد لاحظنا غياب العددين 8 و 9 من أعداد النظام الثماني وأن العـدد 30 في القائمة أعلاه يعني العدد 24 في النظام العشري ، حيث أن كـــل موقـــع فيـــه يساوي ثمانية مرات الموقع الذي يقع على يمينه، سنأخذ عدد الثماني 30 ونطبـــق عليه مبدأ التحويل إلى النظام العشري حيث نضرب العدد 0 بـــ 8^0 والعـــــدد 3 بالعدد 8^1 ثم نجمع الحدين نجد :

$$(30)_8 = 3*8^1 + 0*8^0 = 3*8 + 0*8 = (24)_{10}$$

بعبارة اخرى : لتحويل العدد في النظام الثماني الى النظام العشري نتبع الخطـــوات التالية :

1. نكتب العدد بالشكل الموسع .
2. نوجد قيمة كل حد في الشكل الموسع بالنظام العشري .

3. نوجد حاصل جمع الحدود .

مثال (15) : حول العدد الثماني 3516 الى ما يكافئه في النظام العشري

$$(3516)_8 = 3*8^3+5*8^2+1*8^1+6*8^0 = 3*512 +5*64 + 1*8+6*1$$
$$=1536 + 320 + 8 + 6 = (1870)_{10}$$

أي ان العدد $_8$(3516)يكافئ $_{10}$(1870)، حيث ان العدد 8 هو أساس النظام الثماني والعدد 10 هو أساس النظام العشري .

(3-1-4-3) التحويل من النظام الثماني الى النظام الثنائي وبالعكس

Octal-Binary Interconversion

هناك علاقة قوية بين أساس النظام الثنائي وأساس النظام الثماني وهذه العلاقة هي :

$$8 = 2^3$$

هذا يعني أنه يمكن كتابة كل رقم في النظام الثماني بما يكافئه في النظام الثنائي، وذلك بتبديل كل رقم ثماني بثلاثة ثنائيات، و العكس صحيح، كما في الجدول التالي :

الرقم الثماني OCTAL DIGITS	المكافئات الثنائية BINARY EGUIVALENTS
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

ملاحظة : عند التحويل من النظام الثنائي الى النظام الثماني، نقوم بأخذ ثلاثة أعداد من اليمين الى اليسار ونستبدلها بما يكافئها إذا كان العدد صحيحاً ، أمـــا

إذا كان يحتوي على قسم كسري عندئذ نأخذ كل ثلاثة أعداد إبتداءاً مـن الفاصلة الثنائية التي تفصل القسم الكسري عن القسم الصحيح ونسـتبدلها بما يكافئها في النظام الثماني .

مثال (16) : حول العدد الثماني 3703 الى مكافئه في النظام الثنائي .

الحل : 3 0 7 3

011 000 111 011

و بالتالي نجد $(3702)_8 = (11111000011)_2$

مثال (17) : حول العدد $(10101011111)_2$ الى مكافئه الثماني :

الحل : نقطع العدد الثنائي الى قطع ثلاثية إبتداءاً من اليمين ثم استبدال كل قطعة بما يكافئها في النظام الثماني :

111 011 101 010

7 3 5 2

وبالتالي فإن العدد $(2537)_8$ هو المكافئ الثماني للعدد $(10101011111)_2$

لاحظ أننا أضفنا صفراً للقطعة الأخيرة حتى أصبحت ثلاثية .

مثال (18) : حول العدد $(1101.0101)_2$ إلى مكافئه الثماني .

الحل : نقطع العدد الثنائي الى قطع ثلاثية إبتداءاً من يسار ثم يمين الفاصلة الثنائية ثم نستبدل كل قطعة بمكافئها الثماني .

100 010 . 011 011

4 2 3 3

و بالتالي فإن :

$$(11011.0101)_2 = (33.23)_8$$

(3 – 1 – 4 – 4) العمليات في النظام الثماني :

أولاً: الجمع في النظام الثماني Octal Addition

يتم جمع الأعداد الثمانية وفق الجدول التالي :

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

و لصعوبة حفظ الجدول السابق فإنه يمكن استخدام الأسلوب التالي لإيجاد مجموع عددين في النظام الثماني :

1. نرتب العددين تحت بعضهما البعض إبتداءاً من اليمين .
2. نبدأ من العمود الأول الواقع في الجهة اليمين .
3. نجمع العددين في العمود كما نجمع في النظام العشري .
4. اذا كان الناتج اكبر من (7) نطرح منه العدد (8) ونضع العدد (1) فوق العمود التالي الذي يقع على يساره .
5. نطبق الخطوتين (3 و 4) على العمود التالي مع الانتباه الى انه يجب جمع العدد (1) مع أعداد العمود إذا كان هذا الرقم موجوداً نتيجة الخطوة (4) .

مثال (19) : اجمع كل من التراكيب التالية :

a) $(5)_8+(4)_8$, d) $(7)_8+(4)_8$, g) $(1)_8+(5)_8+(6)_8$

b) $(6)_8+(7)_8$, e) $(1)_8+(4)_8+(2)_8$

c) $(3)_8+(2)_8$, f) $(1)_8+(6)_8+(3)_8$

الحل :

	a	b	c	d	e	f	g
					1	1	1
	5	6	3	7	4	6	5
+	4	7	2	4	2	3	6
	9	13	5	11	7	10	12
-	8	8	0	8	0	8	8
	11	15	5	13	7	12	14

نلاحظ أن المجموع العشري الذي يزيد عن السبعة نطـــرح منـــه العـــدد 8 ونضع على يسار الناتج العدد واحد ، و ان المجموع العشري الذي لا يزيـــد عـــن سبعة نطرح منه العدد صفر ويبقى الناتج على حالة .

مثال (20) : أوجد المجموع الثماني التالي :

$$(7346)_8 + (5263)_8 =$$

الحل :من اجل ايجاد المجموع نرتب العددين بالطريقة العادية ونطبق على كل عمود وبصورة مستقلة قاعدة جمع الأعداد في النظام العشري ، لاحظ أن المحمولات التي تظهر من جراء طرح العدد 8 في الخطوة الخاصة بالتعديل ينقل من اسفل العمود إلى أعلى العمود التالي الواقع إلى اليسار .

	1		1	1	
		7	3	4	6
	+	5	2	6	3
المجموع العشري		12	6	11	9
التعديل		-8	-0	-8	-8
المجموع الثماني	1	4	6	3	1

و بالتالي العدد الناتج هو $(14631)_8$ أي :

$$(7346)_8 + (5263)_8 = (14631)_8$$

ثانياً : الطرح في النظام الثماني Octal Subtraction

لإيجاد الفرق الثماني $y = B - A$ نوجد أولاً المتمم السباعي للرقـــم A (العــدد المطروح) حيث نطرح كل رقم فيه من العدد 7، ثم نجري عملية الجمع العادي التي مرت معنا سابقاً وأخيراً نقوم بحذف الرقم الأخير من الناتج وهو العدد 1 .

مثال(21) : أوجد الفرق الثماني للأعداد $A = 3141$ ، $B = 7526$

حيث $Y = B - A$

الحل : أ. نوجد المتمم السباعي لــ A و ذلك بطرح كل رقم من أرقام A من 7.

متمم A هو : 4636

ت- نجمع متمم A إلى B

	1			1		
		7	5	2	6	B
	+	4	6	3	6	A متمم
الجمع العشري		12	11	6	12	
		-8	-8	-0	-8	
	(1)	4	3	6	4	

ج ــ نحذف العدد (1) الموجود على يسار الناتج نجد الفرق المطلوب

$y = (4364)_8$

(3 – 1 – 5) النظام السادس عشر Hexadecimal System

(3-1-5-1) مقدمة

النظام السادس عشر يستعمل ستة عشر رقماً وهي الأرقام من الصفر حتى التسعة مضافاً إليها ست حروف هجائية وفق ما هو موضح في الجدول التالي :

المكافئات في النظام الثنائي Binary Equivalents	القيم المقابلة في النظام العشري Decimal Values	أرقام النظام السادس عشر HexadecimalDigits
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

3 – 1 – 5 – 2) التحويل من النظام السادس عشر الى النظـــام العشـــري وبالعكس Interconversion Hexadecimal – Decimal

لتحويل العدد في النظام السادس عشر إلى مكافئه في النظام العشري نتبع مايلي :

سوف نوضح ذلك من خلال الامثلة التالية :

1- نكتب العدد السادس عشر بالشكل الموسع .

2- نوجد قيمة لكل حد في الشكل الموسع بالنظام العشري .

3- نوجد حاصل جمع الحدود .

مثال (22) : حول العدد $(73D5)_{16}$ الى مكافئة العشري :

الحل : نعلم ان الحرف D = 13 وبالتالي :

$$(73D5)_{16} = 7*16^3 + 3*16^2 + 13*16^1 + 5*16^0$$
$$= 7*4096 + 3*256 + 13*16 + 5*1$$
$$= 28672 + 768 + 208 + 5 = (29653)_{10}$$

مثال (23) : حول العدد السادس عشر $(39.B8)_{16}$ الى مكافئه العشري .

الحل :

$$(39.B8)_{16} = 3*16^1 + 9*16^0 + 11*16^{-1} + 8*16^{-2}$$
$$= 3*16 + 6*1 + 11* 0.0625 + 8*0.0039062$$
$$= 48 + 9 + 0.6875 + 0.03125 = (57.71875)_{10}$$

لتحويل العدد من النظام العشري الى مكافئه في النظام السادس عشر نتبـــع نفس الخطوات التي استخدمناها لتحويل العدد من النظام العشري الى النظام الثنـائي مع الأخذ بعين الاعتبار أن الأساس هنا هو 16 ، و الأمثلة التالية توضـــح هـــذه العملية.

مثال (24) : حول العدد العشري $P = (9719)_{10}$ الى مكافئه السادس عشر.

الحل : من أجل التحويل من النظام العشري الى النظام السادس عشر نقسم P وكل حاصل قسمة لاحق على الأساس 16 مع ملاحظة البواقي كما في الجدول التالي :

التقسيمات Divisions	ناتج القسمة Quatients	البواقي Remainders
9719 ÷ 16	607	7
607 ÷ 16	37	15
37 ÷ 16	2	5
2 ÷ 16	0	2

ان سلسلة البواقي التي نبدل فيها الباقي العشري 15 بالعدد السادس عشر F ، مقروءة بالترتيب المعكوس ، هي التي تعطي السادس عشر للعدد P=9719 أي أن :

$$(9719)_{10} = (25F7)_{16}$$

مثال (25) : حول الكسر العشري Q = 0.78125 إلى مكافئة السادس عشر .

الحل : من أجل تحويل الكسر العشري الى مكافئة السادس عشر نقوم بضرب الكسر بالأساس 16 حتى تتم عملية تحويل الكسر الى عدد صحيح .

عمليات الضرب Multiplications	الأقسام الصحيحة Integer Parts
0.78125*16=12.50000	12
0.50000*16=8.00000	8

في هذه الحالة نحصل على قسم كسري صفري . إن سلسلة الأقسام الصحيحة والتي نبدل فيها العدد العشري 12 بمكافئه السادس عشر C هي التي تعطي الشكل السادس عشر المطلوب للعدد Q أي :

$$(0.78125)_{10} = (0.C8)_{16}$$

(3-1-5-3) التحويل من النظام السادس عشر الى النظام الثنائي وبالعكس Hexadecimal-Binary Interconversion

هناك علاقة بين أساس النظامين الثنائي والسادس عشر كما يلي :

أي أنه يمكن كتابة كل عدد في النظام السادس عشر بما يكافئه في النظـــام الثنائي، وذلك بتبديل كل عدد ست عشري بأربعة ثنائيات ، والعكس صحيح .

مثال : ما هو المكافئ السادس عشر للعدد الثنائي التالي :

1010001011010010

الحل : نكتب العدد الثنائي على الشكل التالي :

	0010	1101	0010	1010
و هذا يكافئ	2	D	2	A

و بالتالي العدد السادس عشر المكافئ هو $(A2D2)_{16}$

مثال (27) : حول العددين $(DC10)_{16}$ ، $(B747)_{16}$ الى مكافئهما في النظـــام الثنائي .

الحل : نعوض كل رقم من الأعداد السابقة بما يكافئه من الأعداد الثنائية

و بالتالي :

$$(DC10)_{16} = (1101\ 1100\ \ 0001\ 0000)_2$$
$$(B747)_{16} = (1011\ 01110100\ 0111)_2$$

4-5-1-3) العمليات الحسابية في النظام السادس عشرة

أولاً : الجمع في النظام السادس عشر Hexadecimal Addition

يتم جمع الأعداد الستة عشرية وفق الجدول التالي :

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

ولصعوبة حفظ الجدول السابق لاستخدامه عند جمع الأعداد فإنه يمكن استخدام الطريقة التالية :

1. ترتيب العددين المراد جمعهما تحت بعضهما البعض إبتداءاً من اليمين .
2. نبدأ من العمود الأول الواقع جهة اليمين .
3. نجمع الرقمين في العمود كما نجمع في النظام العشري .
4. اذا كان الناتج أكبر من 15 نطرح منه العدد (16) ونضع العدد (1) فوق العمود التالي الذي يقع على يساره .
5. نطبق الخطوتين (3و4) على العمود التالي مع الانتباه إلى أنه يجب جمع العدد (1) مع أرقام العمود إذا كان هذا العدد موجوداً نتيجة الخطوة (4) .

مثال (28) : أوجد ناتج جمع كل من التراكيب التالية في النظام السادس عشر :

a) $(8)_{16}+(9)_{16}$, e) $(C)_{16}+(D)_{16}$, i) $(1)_{16}+(5)_{16}+(C)_{16}$
b) $(3)_{16}+(5)_{16}$, f) $(3)_{16}+(B)_{16}$,j) $(1)_{16}+(E)_{16}+(6)_{16}$

c) $(6)_{16}+(7)_{16}$, g) $(F)_{16}+(D)_{16}$
d) $(A)_{16}+(9)_{16}$, h) $(1)_{16}+(4)_{16}+(6)_{16}$

a	b	c	d	E	f	G	h	i	j
							1	1	1
8	3	6	A	C	3	F	4	5	E
+ 9	+ 5	+ 7	+ 9	+ D	+ B	+D	+ 6	+C	+ 6
17	8	13	19	25	14	28	11	18	21
-16	-0	- 0	-16	-16	-0	-16	-0	-16	-16
11	8	D	13	19	E	1C	B	12	15

مثال (29) : أوجد ناتج جمع ما يلي في النظام السادس عشر :

$(C868)_{16} + (72D9)_{16} =$

الحل : لايجاد المجموع نطبق قاعدة الجمع عمود عمود مع النقل إذا دعت الضرورة الى قمة العمود التالي الواقع الى اليسار مباشرة .

			1	1	
		C	8	6	8
	+	7	2	D	9
المجاميع في النظام العشري		19	11	20	17
التعديلات		-16	-0	-16	-16
المجموع في النظام السادس عشر	1	3	B	4	1

و بالتالي :

$(C868)_{16} + (72D9)_{16} = (13B41)_{16}$

ثانياً : الطرح في النظام السادس عشر Hexadecimal Subtraction

لإيجاد الفرق السادس عشر $Y = L - M$ نوجد أولاً المتمم الخامس عشري للعدد M عن طريق طرح كل رقم في العدد M من 15 ثم إضافة العدد واحد للحصول على المتمم لــ M ، بعد ذلك نقوم بعملية جمع متمم M إلى L .

مثال (30) : أوجد الفرق $Y = L - M$ حيث $L = 72A4$ ، $M = 4E86$.

الحل : نوجد المتمم الخامس عشري للعدد M ، حيث نطرح كل رقم في العدد M من 15 نحصل على B179 وهو المتمم الخمس عشري للعدد M ، نضيف العدد واحد يصبح B17A

الان نضيف متمم M الى L نجد :

	4	A	2	7	
			1		1
L	4	A	2	7	
متمم M	A	7	1	B	+
الجمع العشري	14	17	4	18	
التعديلات	-0	-16	-0	-16	
المجموع السادس عشر	E	1	4	2	1

و بحذف العدد واحد الموجود على يسار الناتج نجد أن

$$Y = 72A4 - 4E86 = 241E$$

3 – 2) المتممات العددية Complements

3–2–1) مفهوم المتمم العددي

إن عبارة متمم تشير الى متمم الأساس ، فاذا كان لدينا العدد 6 فإن متممه العشري هو العدد 4 أي العدد الذي سوف تضيفه اليه حتى يصبح المجموع 10.

تستخدم المتممات في حالتين :

أ. في عملية تخزين الأعداد في الحاسوب ، حيث يوجد هناك العديد من الحواسيب تخزن الأعداد السالبة بإستخدام متمماتها الحسابية .

ب. في عملية الطرح ، حيث يمكن استعمال المتممات لرد عملية الطرح الى عملية جمع ، وهذا الرد مفيد بشكل خاص لتجنب تكرار عملية الاستعارة من عمود الى اخر .

هناك نوعين من المتممات في الأنظمة العددية :

1. المتمم الأساس – ناقص واحد Radix Complement-Minus One
2. المتمم الأساس Radix Complement

سوف نبحث هذين المتممين في النظام العشري وفي النظام الثنائي .

3–2–2) المتممات العشرية في النظام العشري Decimal Complements

هناك نوعان من المتممات للأعداد العشرية ، المتمم الأول يدعى بالمتمم التساعي والمتمم الثاني يدعى بالمتمم العشري .

مثال (31): الجدول التالي يبين المتمم التساعي والعشري لبعض الأعداد في النظام العشري :

العدد العشري	751620	9672	123123	4308
المتمم التساعي	258379	0327	876876	5691
المتمم العشري	248380	0328	876877	5692

حيث حصلنا على المتمم التساعي بطرح كل رقم في العدد المعطى من العدد 9 أما المتمم العشري فهو يساوي الى المتممات التساعية زائد واحد .

مثال (32) : أوجد الفرق $Y = B - A$ حيث $B = 6142$ و $A = 4816$

الحل : أ. نستخدم أولا الطرح العادي :

$$\begin{array}{r} 6142 \\ -\ 4816 \\ \hline 1326 \end{array}$$

نلاحظ أننا يجب أن نستعير مرتين

ب. نجمع المتمم التساعي لـــ A المطروح وهو 5183 إلى B

$$\begin{array}{r} 6142 \\ +\ 5183 \\ \hline (1)\ 1326 \\ 1 \\ \hline 1326 \end{array}$$

هنا اضفنا العدد 1 الى المجموع

جـــ نضيف المتمم العشري لـــ A والذي هو المتمم التساعي زائد واحد أي :

$$5183 + 1 = 5184$$

$$\begin{array}{r} 6142 \\ +\ 5183 \\ \hline (1)\ 1326 \end{array}$$

هنا نهمل العدد 1 للحصول على الحل .

(3-2-3) المتممات في النظام الثنائي **Binary Complements**

يوجد هناك متممات يستخدمان في النظام الثنائي هما المتمم الاحادي والمتمم الثنائي . إذا كان A عدداً ثنائياً ما فإن المتمم الأحادي لـ A نحصل عليه بطرح كل حد من حدود A من العدد واحد ، اما المتمم الثنائي لـ A فهو متممه الأحادي زائد واحد .

مثال (33) : الجدول التالي يبين المتمم الأحادي والثنائي لبعض الأعداد في النظام الثنائي :

111000111000	110100001111	العدد الثنائي
000111000111	001011110000	المتمم الاحادي
000111001000	001011110001	المتمم الثنائي

نلاحظ أننا نحصل على المتمم الأحادي باستبدال الصفر بواحد و الواحد بصفر في النظام الثنائي. نستطيع أن ننجز عملية الطرح في النظام الثنائي بجمع المتمم الأحادي مع العدد واحد أو بجمع المتمم الثنائي.

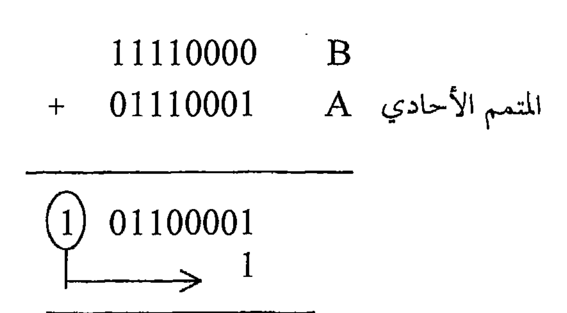
مثال (**34**) : أوجد ناتج $Y = B - A$ حيث $B = 11110000$ و $A = 10001110$

الحل : أ. ننجز عملية الطرح في النظام الثنائي العادية :

$$\begin{array}{rl} 11110000 & B \\ -\,10001110 & A \\ \hline 01100010 & Y \end{array}$$

نلاحظ اننا استعرنا الرقم واحد من الموقع الخامس .

ت. ان المتمم الأحادي لـ A هو 01110001 . نضيف متمم A الى B ثم نجمع واحد



01100010

تدعى هذه الطريقة باسم طريقة النقل الطرفي الالتفافي .

جـ ان المتمم الثنائي لـ A هو 01110010 نضيف هذا المتمم الى B

B	11110000
المتمم الثنائي	1110010 +
	(1) 01100010

نحذف الواحد نحصل على الفرق Y

مثال (35) : أوجد الناتج باستخدام المتمم الثنائي ل : $Y = B - A$

حيث $A = 110011$ و $B = 101010$ ان المتمم الثنائي لـ A هو 001101

، نضيف هذا المتمم الى B نحصل على :

B	101010
متمم A الثنائي	001101 +
Y	110111

نلاحظ انه لا يوجد واحد في الخانة السابقة و بالتالي $Y = 110111$ ليس الفرق المطلوب والسبب هو ان A اكبر من B ، وللحصول على الفرق يجب ان نطرح العدد 1000000 من العدد الناتج وبالتالي الفرق المطلوب هو : (001001 -) .

تمارين وأسئلة عامة

1- حول أعداد النظام الثنائي التالية الى مكافئتها في النظام العشري :

110.011 , 0.10101 , 0.101 , 0.11 , 10010101

2- حول كل من الأعداد التالية من النظام العشري الى النظام الثنائي :

54.321 , 2468 , 0.421875 , 0.3333 ,437.40625

3- أوجد ناتج مايلي :

a) 11011 + 1010 = , b) 110.1101 + 1011.011 =
c) 1101 + 11100 + 1001 + 11001 =

4- أوجد ناتج ما يلي :

a) 110110 * 101 =
b) 111.001 * 1.11 =

5- اوجد قيم عمليات الطرح التالية :

a) 1110001 – 111011 =

b) 1101.0011–110.11011 =

6- أوجد ناتج ما يلي :

111001 ÷ 1001 =

7- حول الأعداد التالية الى مكافئها في النظام الثماني :

a) $(11101010110)_2$
b) $(1001101.01100001)_2$

8- حول العدد العشري $A = 1476$ الى مكافئها في النظام الثماني .

9- حول العدد الثماني $B = 25146_8$ الى مكافئها في النظام العشري .

10- حول الأعداد التالية الى مكافئها في النظام الثنائي :

a) $(430271)_8$ b) $(21.673)_8$

11- حول الأعداد التالية الى مكافئها في النظام الثماني :

a) $(11101010110)_2$, b) $(1001101.01100001)_2$

12- أوجد ناتج ما يلي في النظام الثماني :

a) 3+2 , b) 6+4 , c) 7+5 , d) 2+4 , j) 5+6
i) 7+7 , e) 3+5

13- أوجد ناتج ما يلي :

a) $(6214)_8-(3527)_8$ = , b) $(4617263)_8-(1423736)_8$ =
c) $(6254)_8+(4176)_8$ = , d) $(36517)_8+(64753)_8$ =
e) $(465.37)_8+(31.613)_8$ =

14- حول الكسرين العشريين التاليين الى مكافئها في النظام الثماني :

a) 0.4375 , b) 0.2

15- حول الأعداد العشرية الى مكافئها في النظام السادس عشر :

a) 1000 , b) 15321

16- حول العدد السادس عشر Z = 1A74 الى مكافئها في النظام عشري

17- حول الأعداد السادس عشرة الى مكافئها في النظام الثنائي :

a) 3D59 , b) 27.A3C

18- حول الأعداد الثنائية التالية الى مكافئها في النظام السادس عشر:

a) 10110100101110 , b) 11100.1011011011

19- اجمع الحدود السادس عشرة التالية :

						1
4	8	7	D	3	5	E
+3	+9	+5	+8	+B	+1	+F

20- أوجد ناتج ما يلي :

a) $82C5_{16} + 9D89_{16} =$

21- أوجد المتمم الخمسة عشري و المتمم السادس عشر لكل من :

a) $74B9_{16}$, b) $5C0F8_{16}$, c) $2A7600_{16}$

22- أوجد ناتج ما يلي :

a) $74B64_{16} - 42AF1_{16} =$
b) $9C4D819_{16} - 23C0482_{16} =$

الفصل الرابع

تمثيل البيانات والتعليمات داخل الحاسوب

Data and Instruction Representation

(4 – 1) مفاهيم اساسية

(4–1–1) التمثيل الداخلي **Internal Representation**

كما ذكرنا في الفصل الثاني ، إن وحدة المعالجة المركزية (CPU) تتألف من وحدة التحكم ووحدة الحساب و المنطق ، وكل منها تحتوي على عدد من المسجلات وكل من المسجلات وظيفته الخاصة ، فمسجلات وحدة الحساب والمنطق تقوم بإستيعاب البيانات التي سيتم معالجتها ، أما مسجلات وحدة التحكم فتحتوي على الأوامر و التعليمات التي سيتم تنفيذها .

إن تمثيل البيانات و الأوامر و التعليمات داخل الحاسوب يتم على شكل أرقام ثنائية 1,0 وبالتالي نحتاج لتمثيل حرف واحد الى 8 ثنائيات والتي تسمى بايت Byte ولتمثيل عدد أو جملة ذات معنى يستخدم مفهوم الكلمة Word والذي يتراوح حجمها من 8Bit الى 128 Bit ، حسب نوع الحاسوب وحداثته، وسنبين في هذا الفصل كيفية تمثيل البيانات والتعليمات داخل الحاسوب.

(4 – 1 – 2) العنونة **Addressing**

العنوان بالنسبة للحاسب هو ذلك الرمز الذي يشير الى أوعية التخزين داخل ذاكرة الحاسوب الرئيسية ويميزها عن بعضها البعض ، ويحدد مكان كل منها.

تقسم الذاكرة الرئيسية الى مواقع تخزينية لكل منها عنوان يدل عليها وهذا يختلف حسب نوع الحاسوب ، حيث يمكن التمييز بين نوعين من الحواسيب:

1. حواسيب معنونة الكلمة **Word Addressable Machine**

في هذا النوع من الحواسيب يتم تمثيل البيانات العددية والتعليمات على شكل كلمات ، أما البيانات غير العددية كالحروف والعلامات الخاصة فيتم تمثيل كل حرف في بايت Byte وتجمع عدد من البايتات معاً لتشكل كلمة معنونة واحدة، أي أن العنوان يكون للكلمة ، لذلك فالحواسيب معنونة الكلمة تكون أنسب للتطبيقات العلمية والرياضية ولغاتها .

2. حواسيب معنونة البايت **Byte Addressable Machine**

حيث تنقسم الذاكرة إلى مجموعة من البايت كل منها له عنوان أي أن العنوان هنا للبايت وليس للكلمة، يتم تمثيل البيانات التي تحتاج الى كلمة كاملة مثل البيانات العددية عن طريق تجميع عدد من البايتات متجاورة معاً والتعامل معها كوحدة واحدة عنوانها هو عنوان البايت الأولى، هذا النوع من الحواسيب يكون انسب للتطبيقات التي لا تحتاج عمليات حسابية كثيرة وكذلك التطبيقات التي تعتمد على البيانات غير العددية .

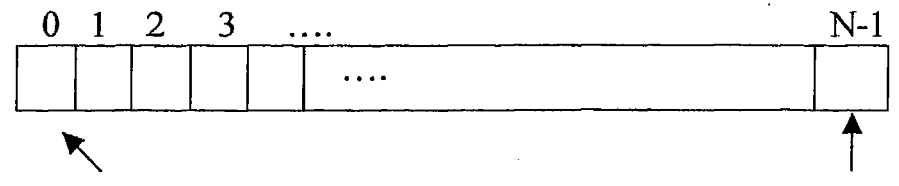
(4 – 2) تمثيل الأعداد الصحيحة :

Representation Integer Number:

المقصود بالأعداد الصحيحة، تلك القيم العددية التي لا تحتوي على فاصلة عشرية (نقطة كسرية)، هذه العلامة الكسرية يفترض وجودها دائماً في موقع ثابت في أقصى اليمين من العدد بدون ان تظهر، فالعدد يكون صحيحا فقط إذا خلا من النقطة الكسرية الظاهرة ، هناك اكثر من طريقة لتمثيل القيم العددية الصحيحة داخل الحاسوب .

(1–2–4) تمثيل العدد وإشارته :

وفقا لهذه الطريقة يتم تمثيل القيم الصحيحة داخل الحاسوب في مسجل ينقسم الى N من المواقع الثنائية مرقمة من (0) الى (N − 1) .



العلامة الكسرية الافتراضية   موقـــع الإشـــــارة

نلاحظ من الشكل السابق انه يخصص اقصى موقع ثنائي في اليسار لتمثيل الإشـــارة فيوضع العدد (0) لتمثيل الاشارة الموجبة (+) أو العدد (1) لتمثيل الإشـــارة السالبة (−) ، أما بقية مواقع المسجل فيخصص لتمثيل العدد الصحيح وذلك مـــن اليمين الى اليسار ، العلامة الكسرية يفترض وجودها في اقصى اليمـــين ولا تـــأخذ موقعاً.

مثال (1) : الجدول التالي يعرض لنا امثلة ايضاحية لكيفية تمثيل الأعداد الصحيحـــة سالبة أو موجبة وذلك في مسجل يتألف من ثمانية ثنائيات أي 8Bit .

مثال	العدد الثنائي	تمثيله في مسجل طوله 8 bit							
1	110110	0	0	1	1	0	1	1	0
2	- 10110	1	0	0	1	0	1	1	0
3	10100	0	0	0	1	0	1	0	0
4	- 100	1	0	0	0	0	1	0	0
5	- 1111010	1	1	1	1	1	0	1	0
6	111100101	0	1	1	0	0	1	0	1

من الأمثلة السابقة نستنتج ما يلي :

1. يخصص اقصى موقع من اليسار لتمثيل الاشارة .
2. العلامة الكسرية يفترض وجودها في أقصى اليمين و لا تأخذ موقعاً .
3. تمثيل العدد الصحيح يكون من اليمين الى اليسار .

4. اذا كان عدد مواقع العدد أقل من عدد مواقع المسجل ، تملأ بقية مواقع المسجل بالأصفار دون التعرض لموقع قيمة الإشارة .

5. اذا كان عدد مواقع العدد اكبر من عدد مواقع المسجل، فان اعداد اليسار الـتي تزيد عن حجم المسجل تحذف دون التعرض لموقع وقيمة الإشارة ، ففي المثـال (1) البند السادس يتم تمثيل السبعة أرقام الأولى من اليمين إلى اليسار ويتـم حـذف الرقمين الأخيرين (11) مع ملاحظة أن لهما اكبر قيمة في العدد .

(4-2-2) التمثيل بالمتمم الثنائي :

في هذه الطريقة تمثل الإشارة ومتمم العدد، سواء كان ذلك للقيم الموجبـة أو السالبة :

1. تمثيل القيم الموجبة بالمتمم الثنائي : متمم القيمة الموجبة هو القيمة نفسها . فلـو أردنا تمثيل القيمة 1001101 في مسجل ذو ثمانية مواقع فان ذلك يكـون على الشكل التالي :

0	1	0	0	1	1	0	1

2. تمثيل القيم السالبة : ليكن لدينا القيمة (51 -) ولنرى خطوات تمثيل هـذه القيمة :

القيمة (51 -) تعادل بالنظام الثنائي	:	-110011
باستخدام مسجل ذو ثمانية مواقع يكون	:	0110011 -
و بالحصول على المتمم الثنائي تصبح	:	11001101

أي تمثل على الشكل التالي :

1	1	0	0	1	1	0	1

(4-2-3) حدود التمثيل ومشكلة الفائض :

يتم تخزين (تمثيل) القيم الصحيحة ضمن حدود معينة تتوقف على طـــول المسجل وذلك كالآتي :

أ. اكبر قيمة صحيحة موجبة يمكن تمثيلها وفق نظام النقطة الثابتة نحصل عليها مـــن العلاقة ($2^{n-1} - 1$) حيث تمثل N عدد المواقع الثنائية في المسجل .

فاذا كان لدينا مسجل بطول ثمانية مواقع ثنائية فان اكبر قيمة يمكن تخزينـــها في المسجل هي :

$$(2^{8-1} - 1) = 2^7 - 1 = 127$$

وهذه القيمة تمثل كالتالي :

0	1	1	1	1	1	1	1

أي اننا لا نستطيع تمثيل القيمة 128 + أو أي قيمة اكبر منها في هـــذا المســـجل يحوي 8 مواقع (8Bit) .

ب. اقل قيمة صحيحة يمكن تمثيلها في المسجل السابق هي :

$$(- 2^{8-1} + 1) = - 2^7 + 1 = - 128 + 1 = - 127$$

أي اننا لا نستطيع تمثيل القيمة 128 – أو اقل منها في هذا المسجل ، وبالتـــالي نحصل على تمثيل خاطئ عندما نريد تمثيل قيمة اكبر أو اصغر من المجال (+127 , -127) .

من الجدير بالذكر أن الحاسبات الحديثة تستخدم مسجلات بسعة 32 موقع ثنائي ومثل هذه المسجلات تكون قادرة على استيعاب قيم حـــتى 2147483647 أي ($2^{32-1} - 1$) وبالتالي يكفي أن نعرف طول المسجل في الحاسوب لمعرفـــة العـــدد الممكن تخزينه .

(4 – 3) تمثيل الأعداد ذات النقطة العائمة

Floating Point Representation

و هي اكثر طرق تمثيل الأعداد أهمية وذلك لأنها تتميز بما يلي :

1. **امكانية** تمثيل الأعداد الكبيرة والصغيرة التي لا يمكن تمثيلها باستخدام طريقة النقطة الثابتة .
2. **الحصول** على درجة دقة عالية جداً بالمقارنة بأسلوب النقطة الثابتة خاصة بالنسبة للجزء الكسري .
3. **تعتبر** أنسب الطرق لتمثيل الأعداد الكسرية أو التي تحتوي على جزء صحيح وآخر كسري .
4. **يمكن** زيادة درجة الدقة في التمثيل عن طريق ما يسمى بالدقة المضاعفة .
5. **يمكن** بواسطتها تمثيل الأعداد المركبة بجزئيها الحقيقي والتخيلي .

لإيضاح فكرة الأعداد ذات النقطة العائمة وتمثيلها ، نفترض ان لدينا عددين الأول هو 1200000000 والثاني هو 0.000000009 فإننا لا نستطيع تمثيلها داخل الحاسوب باستخدام أسلوب النقطة الثابتة ، وذلك لكبر الأول وصغر الثاني ,ولكننا اذا أعدنا صياغة العددين بصورة كسرية أسيه يصبحان كالتالي :

$$12000\,000\,000 = 0.12 \times 10^{11}$$
$$0.000\,000\,00\,9 = 0.9 \times 10^{-8}$$

أي اننا نحصل على صيغة رقمية ممكنة التمثيل . هذه الصيغة العددية هي التي تسمى بشكل الأعداد عائمة النقطة وصيغتها العامة تكون على الشكل التالي :

$$\pm M * R^{E}$$

حيث ان :

M – تعبر عن الصيغة الكسرية للعدد و تسمى القاعدة .

R – تعبر عن اساس النظام المستخدم

E – تعبر عن الأس المرفوع اليه الاساس

وهذا يعني انه لتمثيل أي رقم وفق نظام النقطة العائمة ، ينبغـــــي أولاً وصفـــه في الصيغة السابقة، والتي تسمى بالشكل المعياري و من ثم تخزينـــه في مســـجل ذي مواصفات تتفق وهذه الصيغة.

(1-3-4) الشكل المعياري Standard From

عند تمثيل العدد باستخدام صيغة الأعداد عائمة النقطة يجب الأخذ بعـــين الاعتبار ما يلي :

1. العلامة الكسرية تكون دائماً في أقصى اليسار من العدد مثل القاعدة للعـــدد العشري 0.1275 أو القاعدة للعدد الثنائية 0.101011 .
2. من الخطأ ان تكون القاعدة مثلاً على الشكل التالي 0.000234

والصحيح يجب ألا يكون هناك اصفار على يمين العلامة الكسرية ، وإن وجد يجب تحويل العدد إلى صيغة تحذف بها الأصفار ، أي الشكل :

$$0.234 * 10^{-3}$$

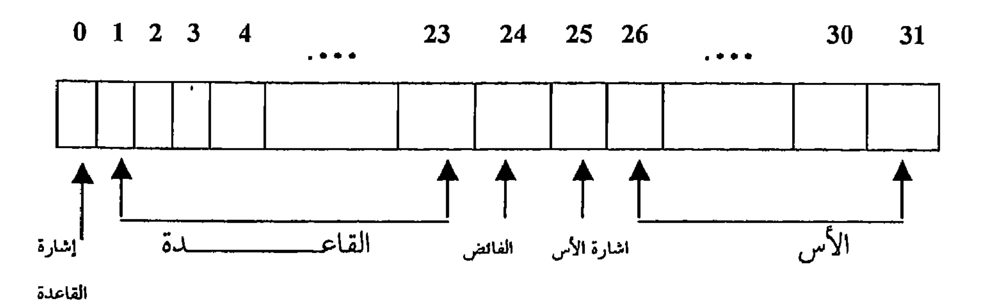
مثال (2) : الجدول التالي يبين بعض الامثلة لتحويل الأعــــداد مـــن الصيغـــة المستعارة إلى الصيغة المعيارية .

الشكل المعياري	العدد
$0.4234 * 10^4$	4234
$0.12342 * 10^3$	123.42
$- 0.1234 * 10^1$	- 1.234
$0.51 * 10^{-3}$	0.00051
$- 0.5 * 10^{-1}$	- 0.050
$0.110101 * 2^{+4}$	1101.01
$0.1011 * 2^2$	10.1101
$- 0.1011 * 2^0$	- 0.1011
$0.11 * 2^{-5}$	0.0000011

(2-3-4) المسجل ذو النقطة العائمة

Floating Point Register

يتم تمثيل الأعداد بعد تحويلها إلى الشكل المعياري في مسجل خاص ويختلف هذا المسجل من نظام الى نظام اخر سواء في طوله أو في شكله ، على سبيل المثال اذا كان طول المسجل هو 32 بت فانه يمكن تصور الشكل التالي للمسجل ذي النقطة العائمة :



حيث ان :

1. يتم تخصيص موقع واحد لاشارة القاعدة واخر لاشارة الاس .
2. يتم تخصيص عدة مواقع (23 موقعا هنا) لتمثيل القاعدة .
3. تخصص عدة مواقع لتمثيل الأس .
4. يخصص موقع واحد للفائض حيث يحتوي على العدد 1 إذا كانت قيمة الاس أكبر من المواقع المخصصة لتخزينه، وتحتوي على القيمة 0 إذا كانت قيمة الأس في حدود تلك المواقع .

مثال (3) : ليكن لدينا مسجلا ذو 12 موقعا ثنائي، ستة منها للقاعدة وثلاثة للأس وموقع واحد لكل من إشارة القاعدة وإشارة الأس والفائض .

1- كيف يتم تمثيل العدد العشري 7.5 .

الحل : لتمثيل العدد 7.5 نتبع الخطوات التالية :

أ. نحصل على المكافئ الثنائي للرقم و هو 111.1

ب. نحوله الى الشكل المعياري فيكون $0.1111* 2^3$

ج ــ نمثل الصيغة المعيارية في المسجل ذي النقطة العائمة كالتالي :

0	1	1	1	1	0	0	0	0	0	1	1
اشارة القاعدة			القاعدة				الفائض	إشارة الأس		الأس	

2 - كيف يتم تمثيل العدد 11.625

الحل : العدد الثنائي المكافئ هو 1011.101 ، نحول هـــذا العـــدد الى الشـــكل المعياري فيكون $0.1011101 * 2^4$ و بالتالي يمثل على الشكل التالي :

0	1	0	1	1	1	0	0	0	1	0	0

نلاحظ ضياع آخر عدد من يمين قاعدة العدد وذلك بسبب عدم كفايـــــة الجـــزء المخصص للقاعدة

3- العدد العشري 131.5 يناظرها العدد 10000011.1 شـــكلة المعيـــاري $0.100000111 * 2^8$ وبالتالي يتم تمثيله على الشكل التالي:

0	1 0 0 0 0 0	1	0	0 0 0

نلاحظ ان قيمة الأس اكبر من سعة الجزء المخصص له وبالتالي ظـــهر العـــدد1 في موقع الفائض . كما نلاحظ أنه تم اهمال 111 من اليمين في القاعدة لعدم اتســـاع الجزء المخصص لها .

4- القيمة الثنائية 0.000001 يعبر عنها معيارياً $0.1* 2^{-5}$ و يمثل داخل المسجل كما يلي :

0	1	0	0	0	0	0	0	1	1	0	1

نلاحظ أن الأس السالب قد تم تمثيله في صورة متممة الثنائي .

ملاحظة : لتمثيل الأعداد بدقة مضاعفة نستخدم مسجلين ، كما يمكن تخزين القيم المركبة داخل مسجلين بحيث يخزن الجزء الحقيقي لهذه القيم داخل مسجل والجزء التخيلي داخل المسجل الآخر .

(4 – 4) تمثيل الأحرف والرموز

Character Representation

المقصود بالأحرف و الرموز ما يلي :

- الاعداد التي لا تدخل في العمليات الحسابية (9 , ... 1 , 0) والتي تعامل كرموز وليس كاعداد .
- الحروف الهجائية (Z , ... , A) و (a, ...,z) .
- الرموز و العلامات الخاصة (... , s ( ) " ; , # \ / * ... .)

يتم تمثيل هذه الاحرف والرموز في وحدة تخزينية تسمى بايت والبايت كما ذكرنا تتكون من ثمانية مواقع ثنائية . على الرغم من تنوع الأنظمة المستخدمة في تمثيل الحروف إلا أنها تعتمد جميعها على تقسيم البايت إلى جزئين .

أ. قسم النطاق Zone Area : ويتكون من أربعة مواقع ثنائية أو من موقعين ثنائيين حسب النظام المستخدم .

ب. القسم العددي Numeric Area ويتكون من أربعة مواقع ثنائية .

يوجد هناك أنظمة لتمثيل الحروف و الرموز اشهرها :

(1–4–4) نظام Binary Coded Decimal (BCD)

استخدمت هذه الشفرة في الاجيال المبكرة للحواسيب ويندر استخدامها في الوقت الحالي ، حيث تمثل الحروف والرموز والأعداد في بايت ذات ست مواقع، والبايت في هذا النظام يستوعب عددين ثمانيين (العدد الثماني يمثل في ثلاثة مواقع)، حيث قسم النطاق يتكون من موقعين ، انظر الجدول (1) .

مثال (4) : تمثل كلمة (Ali) في ثلاث بايتات كالاتي :

A					L								I				
1	1	0	0	0	1	1	0	0	0	1	1	1	1	1	0	0	1

(4-4-2) نظام EBCDIC

Extended Binary Coded Decimal Interchange Code

يعتمد هذا النظام على البايت ذات الثمانية مواقع، أربعة للنطاق و أربعة للقطاع العددي، والبايت يستوعب عددين ست عشريين أنظر الجدول (1)

مثال : تمثل كلمة (TO) في نظام EBCDIC كآلاتي :

T								O							
1	0	1	1	0	1	0	0	1	0	1	0	1	1	1	0

(4-4-3) نظام (ASCII)

American Standard Code for Information Interchange

يعتمد هذا النظام على البايت ذات ثمانية مواقع ، أربعة للنطاق وأربعة للقطاع العددي ، انظر الجدول (1) .

مثال : تمثل الكلمة (TO) في نظام ASCII كالآتي :

T								O							
1	0	1	1	0	1	0	0	1	0	1	0	1	1	1	1

الجدول (1)

الحرف	BCD	EBCDIC	ASCII
A	11 0001	1100 0001	1010 0001
B	11 0010	1100 0010	1010 0010
C	11 0011	1100 0011	1010 0011
D	11 0100	1100 0100	1010 0100
E	11 0101	1100 0101	1010 0101
F	11 0110	1100 0110	1010 0110
G	11 0111	1100 0111	1010 0111
H	11 1000	1100 1000	1010 1000
I	11 1001	1100 1001	1010 1001
J	10 0001	1101 0001	1010 1010
K	10 0010	1101 0010	1010 1011
L	10 0011	1101 0011	1010 1100
M	10 0100	1101 0100	1010 1101
N	10 0101	1101 0101	1010 1110
O	10 0110	1101 0110	1010 1111
P	10 0111	1101 0111	1011 0000
Q	10 1000	1101 1000	1011 0001
R	10 1001	1101 1001	1011 0010
S	01 0010	1110 0010	1011 0011
T	01 0011	1110 0011	1011 0100
U	01 0100	1110 0100	1011 0101
V	01 0101	1110 0101	1011 0110
W	01 0110	1110 0110	1011 0111
X	01 0111	1110 0111	1011 1000
Y	01 1000	1110 1000	1011 1001
Z	01 1001	1110 1001	1011 1010
0	00 1010	1111 0000	0101 0000
1	00 0001	1111 0001	0101 0001
2	00 0010	1111 0010	0101 0010
3	00 0011	1111 0011	0101 0011
4	00 0100	1111 0100	0101 0100
5	00 0101	1111 0101	0101 0101
6	00 0110	1111 0110	0101 0110
7	00 0111	1111 0111	0101 0111
8	00 1000	1111 1000	0101 1000
9	00 1001	1111 1001	0101 1001

(4 – 5) تمثيل الأوامر والتعليمات

من اجل التعرف على كيفية تمثيل الأوامر والتعليمات يجب التعرف على مفاهيم التعليمة أو الأمر والبرنامج ثم ننتقل الى الكيفية التي يتم بها تمثيلها في ذاكرة الحاسوب .

التعليمة Instruction أو الأمر Order هو ايعاز للحاسوب لانجاز عملية معينة مثلا :

التعليمة Z=A+B إيعاز للحاسوب بجمع قيمة A الى B وتخزينها في Z

التعليمة Read (x) إيعاز للحاسوب بإدخال قيمة X للذاكرة .

التعليمة Write (y) إيعاز للحاسوب بإخراج قيمة y كنتائج مطبوعة .

كل مجموعة مترابطة منطقياً من التعليمات والأوامر والمتعلقة بمعالجة مشكلة محددة تسمى برنامجاً Program .

إن عملية معالجة البرنامج تتطلب أن يكون البرنامج مخزناً في الذاكرة ويتم التعامل معه تعليمة ، تعليمة بالتسلسل داخل وحدة المعالجة المركزية .

يحتل الأمر الواحد في ذاكرة الحاسوب كلمة واحدة ، وهذا الأمر أو التعليمة ينقسم الى مقطعين :

1. **مقطع رمز العملية** (Operation Code) والخاص بتوريد العملية التي يراد من الحاسوب انجازها مثل ADD أو LOAD وبعبارة أخرى هو رمز فعل الأمر في هذه التعليمة .
2. **مقطع العنوان** (Address Section) والخاص بتحديد عنوان أو عناوين الوحدات التخزينية التي ستخضع لعملية المعالجة ، وقد يسمى Operand Field .

فعلى سبيل المثال اذا كان لدينا الأمر التالي بلغة الحاسوب Assembly

Add 542

والذي يعني جمع محتويات المخزن الذي عنوانه 542 تمثل عنوان المخـــزن الــذي يحتوي على البيانات المراد جمعها الى المراكم ، وبالتالي فان الكلمة أو المسجل الذي سيستوعب هذا الأمر سيكون كالتالي :

ADD	542

لنفترض ان طول الكلمة التي ستستوعب الأمر هو (16 bit) منها اربعة لرمـــز العملية والباقي للعنوان ، ولنفترض ايضا ان المكافئ الثنائي لعملية الجمع هـو 0101 فإننا نستطيع تمثيل الأمر السابق في الكلمة ذات الستة عشر موقعا ثنائيا كالتالي :

0	1	0	1	0	0	1	0	0	0	0	1	1	1	1	0

مع ملاحظة ان : $(542)_{10} = (1011011100)_2$

وليس هذا هو الشكل الوحيد لتمثيل الأوامر إذ أنه يمكن التمييز بـــين الاشـــكال الثلاثة التالية:

أ. **الأمر أحادى العنوان** وقد تم عرضه اعلاه .

ب. **الأمر ثنائي العنوان** ،حيث يكون هناك عنوانان في الأمر الواحد ، فمثلا :

ADD X , y

والذي يعني جمع القيمة الموجودة في المخزن الذي عنوانه X الى القيمة الموجـودة في المخزن الذي عنوانه Y وتخزين الناتج في المخزن الثاني :

						X						Y			
0	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1

على اعتبار ان عنوان المخزن الأول والذي تمثله X هو 55 و عنـــوان المخـــزن الثاني والذي تمثله Y هو 13 .

ج ـــ **الأمر ثلاثي العنوان** ، يحتوي الأمر على ثلاثة عناوين ويتم تمثيلـــه بنفـــس الأسلوب السابق .

فمثلاً الأمر :

$$Z = X + Y$$

و الذي يعني جمع قيمة X و Y وتخزينها في المخزن الذي عنوانه Z لو اعتبرنا أن عنوان المخزن الأول 33 و عنوان المخزن Y هو 3 و عنوان المخزن Z هو 36 فيكون تمثيله على الشكل التالي :

						X						Y						Z			
0	1	0	1	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	0	0

أسئلة وتمارين عامة

1. كيف يتم تمثيل الأعداد الثنائية التالية في مسجل طوله 8 بت ؟

أ. 100110011 ب. 10110 ج. 10100111

2. كيف يتم تمثيل الأعداد الثنائية التالية في مسجل طولة 12بت

أ. 462 - ب. 1357 ج. 63 –

3. استخدام التمثيل بالمتمم الثنائي للأعداد التالية في مسجل ذو ثمانية مواقع :

أ. 101010101 ب. 111000 ج. 98

د. 112 – هـ. 135 و. 111111

4. اكتب الشكل المعياري للاعداد التالية :

أ. 65321 ب. 632.145 ج. 22.34 –

د. 0.00101 هـ. 0.000111 – و. 11011.1011

5. ليكن لدينا مسجلا ذو 12 موقعاً ثنائياً، ستة منها للقاعدة ،كيف يتم تمثيل الاعداد التالية :

أ. 18.5 – ب. 132.5 ج. 6 –

د. 0.100001 هـ. 1101.0011 و. 1111.10011 -

الفصل الخامس

الكيان المنطقي للحاسوب (البرمجيات)

SOFTWARE

(5 – 1) ماهية البرمجيات :

من المعروف أن نظام الحاسوب يتكون من جزأين رئيسيين :

1. المكونات المادية Hardware والتي سبق شرحها في الفصل الثاني .
2. برمجيات الحاسوب Software .

و البرنامج هو عبارة عن مجموعة من التعليمات تكتب بلغة معينة وتخزن في الذاكر لتخبر الحاسوب ماذا وكيف يعمل لتنفيذ مهمة معينة وتمكنه من إدارة ومراقبة وتنظيم مكوناته المادية لتحقيق تلك المهمة ، بعبارة أخرى برامج الحاسوب بمثابة الروح من الجسد وإن أي نظام حاسوبي بدونها يصبح عديم الجدوى.

(5 – 2) أنواع برمجيات الحاسوب

هناك نوعان رئيسيان من برمجيات الحاسوب يطلق على إحداها برمجيات النظام System Software والنوع الآخر برمجيات التطبيقات Application Software.

(5–2–1) برمجيات النظام : System Software

وهي البرامج المعدة بواسطة الجهة الصانعة للحواسيب والتي يستخدمها الحاسوب ليتحكم ويوجه ويشرف على نظام الحاسوب بأكمله من مكوناته المادية وبرامج التطبيقات ، وهذه البرامج إما أن تكون مخزنة تخزيناً دائماً في ذاكرة القراءة

فقط ROM وبعضها الآخر يكون مخزنا على وسيط خارجي (أقـــراص ممغنطـــة) تستدعى وقت الحاجة إلى الذاكرة المؤقتة RAM .

ويمكن تقسيم برمجيات النظام إلى ثلاثة مجموعات أساسية هي :

أولاً : أنظمة التشغيل Operating Systems

وهي الأنظمة والبرامج المسؤولة عن عمليات تشغيل الأجهزة والمعـــــدات وفق نظام خاص وخطوات مرتبة من أجل معالجة البيانات والإشراف على تشـــغيل جميع برامج التطبيقات ، وهناك العديد من أنظمة التشغيل المستخدمة والمتاحة والتي يتم أعدادها بواسطة منتجي الحواسيب ونظام التشغيل MS-DOS يعتبر اكـــثر نظم التشغيل شهرة للتعامل مع الحواسيب الصغيرة ، ونظام التشغيل Unix يعتـــبر من أنظمة التشغيل الشهيرة للحواسيب المتوسطة ..(وسنتناول موضـــوع أنظمـــة التشغيل بالتفصيل لاحقا إن شاء الله) .

ثانياً : مترجمات اللغات Language Compilers

وهي البرامج المسؤولة عن تحويل أوامر وإيعازات البرنـــامج الى إشـــارات يستطيع الحاسوب تفسيرها والعمل على أساسها ، أي إلى لغة الآلة والتي تتـــألف بدورها من تكوينات من النظام الثنائي الصفر والواحد ، ويطلق علـــى البرنـــامج المكتوب بلغة عالية المستوى بالبرنامج المصدر أو (رمز المصدر Source Code)، ويطلق على البرنامج المكتوب بلغة الآلة البرنامج الهدف أو (رمز الهـدف Object Code) .

وهناك نوع آخر من برامج الترجمة يطلق عليها اسم المفســر Interpreter والفرق بين المترجم والمفسر هو أن المترجم أسرع من المفسر بكثـــير وذلـــك لان المترجم يترجم البرنامج المصدر مرة واحدة ثم يقوم بتخزين برنامج الهـــدف برمـــز الآلة كملف Object Code File وذلك لاستخدامه عند الحاجة دون أن يترجم

البرنامج المصدر مرة أخرى ، أما المفسر فإنه يقوم بترجمة البرنامج المصدر خطوة خطوة أثناء تنفيذ البرنامج ولذلك فان البرنامج المترجم يحتاج حيز في الذاكرة أكبر من ذلك الذي يحتاجه البرنامج المفسر ، كما أن البرنامج المفسر يمكن إيقافه أثناء التشغيل عند أي موضع ثم إعادة تشغيله مرة أخرى وذلك بعكس البرنامج المترجم.

ثالثاً : البرنامج المساعدة Utilities

وهو عبارة عن مجموعة من البرامج المنفصلة التي تؤدي كل منها وظيفة أو وظائف محددة ، تستخدم أساسا للسيطرة والتحكم في مكونات الحاسوب المادية وغير المادية بالإضافة إلى تنفيذ وظائف نظام التشغيل بشكل أسهل ، ومن أمثلة هذه البرامج أدوات الحاسوب الشخصي PC-Tolls وبرامج أدوات نورتون Norton Utilities .

(5-2-2) برمجيات التطبيقات Application Software

وهي تلك البرامج التي تعد من قبل مستخدمي الحاسوب من المبرمجين بهدف حل مشكلة معينة أو مسالة معينة تواجههم ، وذلك فإن هذا النوع من البرامج تشتمل تقريباً جميع أنشطة الحياة ، ويمكن لمستخدمي الحاسوب الحصول على برامج التطبيقات عن طريقين :

1. أن يقوم المستخدم بكتابة البرامج بنفسه في العادة يستخدم لذلك إحدى اللغات العالية المستوى (وسنتناول ذلك في الفصلين السابع و الثامن) .
2. أن يقوم المستخدم بشراء برامج جاهزة خططت و كتبت بواسطة مجموعة من المبرجين المتخصصين ، ويوجد حالياً الآلاف من هذه البرامج وفي جميع المجالات المختلفة ، وهنا لا بد من الإشارة إلى أن برامج التطبيقات تكتب لنظام تشغيل معين أي أن برامج التطبيقات المكتوبة لنظام تشغيل محدد لا يمكن تنفيذها باستخدام نظام تشغيل أخر بدون أجراء تعديلات ، وحتى إذا كانت الحواسيب المختلفة تشترك في نفس نظام التشغيل ولكنها مصنعة بواسطة

منتجين مختلفين فانه لا بد من إجراء بعض التعديلات على البرامج ، ولذلك فمن المهم للمستخدم قبل شراء أي حزمة من البرامج التطبيقية معرفة نوع الحاسوب الذي يعمل عليه ونوع نظام التشغيل المستخدم .

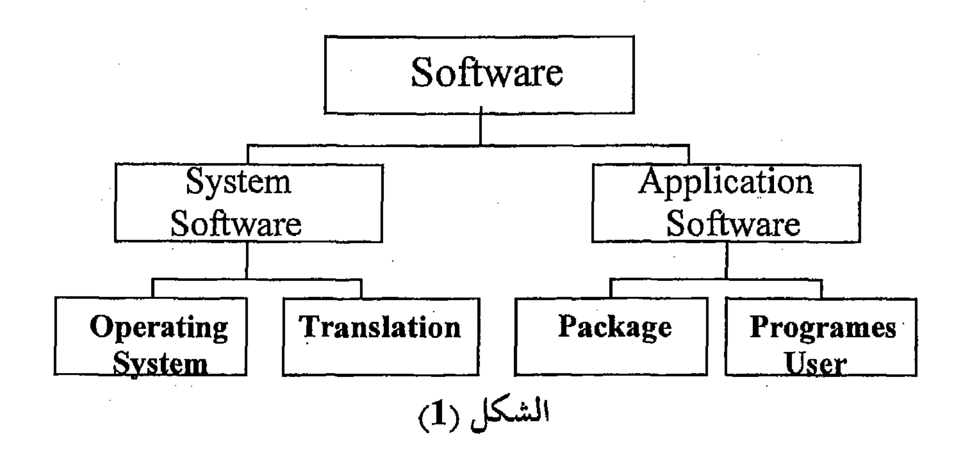
وبشكل عام تقسم برمجيات التطبيقات الى نوعين :

أولاً : برمجيات التطبيقات الجاهزة :

وتدعى أيضاً الحزم البرمجية(Packages) ومن أمثلة تلك البرامج برامج معالجة النصوص Word Processors وبرامج الجداول الإلكترونية Spread Sheets ، وبرامج نظم ادارة قواعد البيانات Data Base Management Systems ، وغيرها ... ، وهذه النوعية من البرامج يكتبها مبرمجون متخصصون وسنلقي الضوء عليها في الفقرات اللاحقة ان شاء الله.

ثانياً : برمجيات التطبيقات الخاصة :

وهي برامج لحل مسائل أو مشاكل خاصة يعدها مستخدموا الحاسوب وكل مستخدم في مجاله مثل البرامج الخاصة لحل المسائل الرياضية المحددة وبرامج لحل مسائل النقل والبرمجة الخطية وبرامج تحليل البيانات الإحصائية وإجراء الاختبارات الإحصائية، وبرامج الأعمال خاصة بالشركات والهيئات ... والشكل (1) يلخص الأقسام المختلفة لبرمجيات الحاسوب .



الشكل (1)

(5 – 3) نظام التشغيل Operating System

(5-3-1) تعريف نظام التشغيل

إن التقدم العلمي والتكنولوجي الذي يشهده العالم اليوم في إنتاج الحواسيب الإلكترونية وتطبيقاتها الواسعة في شتى الميادين ، أدى إلى وجوب معرفة ودراسة نظم التشغيل المختلفة التي تساعد المستخدم على التعامل مع الأنواع المختلفة من الحواسيب ، حيث أن اختيار الحاسوب المناسب لا يعتمد فقط على المكونات المادية Hardware وعلى اللغات العالية المستوى والبرامج التطبيقية التي تتعامل مع هذه المكونات بل يعتمد بصورة أساسية على نظم التشغيل المستخدمة والتي تتعامل مع تلك المكونات .

نظام التشغيل يمثل حلقة الاتصال بين المستخدم والحاسوب ، فالمستخدم يتعامل مع مكونات الحاسوب المختلفة من خلال نظام التشغيل ، حيث أنه ينظم العمليات التي يقوم بها الحاسوب .

ويمكن تشبيه نظام التشغيل بالمدير الذي يدير مشروعا معيناً ولكي يكتب لهذا المشروع النجاح وبكفاءة عالية فإن على المدير أن يقوم بتنظيم أنشطة المشروع والموارد المستخدمة في تنفيذ هذه الأنشطة بهدف تنفيذ المشروع في أقل وقت ممكن وبأكفأ استغلال للموارد المتاحة . حيث أن نظام التشغيل يقوم بنفس العملية ، مع الأخذ بعين الاعتبار أن الموارد في هذه الحالة هي مكونات الحاسوب والتي تشمل الوحدات التالية :

وحدة المعالجة المركزية	Central Processing Unit
الذاكرة الرئيسية	Main Memory
وحدات الإدخال والإخراج	Input/Output Units

كما أن هذه الموارد تشتمل أيضاً على الموارد البشرية المتمثلة بمشغل الحاسوب والمبرمج ومحلل النظم ..الخ

أما الأنشطة فهي جميع العمليات التي يقوم بها الحاسوب من تحميل للبرامج وإدخال للبيانات ومعالجة لهذه البيانات .. الخ

من هذه المقدمة البسيطة يتضح أن نظام التشغيل يمثل العمود الفقري لنظام الحاسوب والذي بدونه يصعب تعامل الإنسان مع الحاسوب والاستفادة من إمكانياته .

(5-3-2) الخصائص العامة لنظام التشغيل :

في الوقت الحاضر تتنافس نظم التشغيل فيما بينها للوصول إلى الصورة المثالية لاستخدام الحاسوب ومكوناته ، ونظام التشغيل المثالي يتمتع بالخصائص التالية :

1. التحكم والسيطرة على مكونات الحاسوب وذلك بالربط بين وحدة التشغيل المركزية التي تعتبر بمثابة عقل الحاسوب وبين باقي المكونات .
2. أداء العمليات الأساسية التي تساعد المستخدم على التعامل مع مكونات الحاسوب مثل نسخ الملفات ومسح ونسخ الأقراص وتشكيلها .. الخ وذلك باستخدام مجموعة من الأوامر Commands يسهل على المستخدم حفظها، والتعامل مع الحاسوب من خلالها.
3. التعامل مع برامج التطبيقات Application Programs مثل الجداول الإلكترونية وبرامج معالجة الكلمات ونظم إدارة قواعد البيانات .
4. تنظيم الأعمال التي يقوم بها الحاسوب Job Control .
5. القدرة على أداء عدة وظائف في نفس الوقت Multitasking .
6. السماح لعدد من المستخدمين بالتعامل مع الحاسوب في وقت واحد Multiusers .

7. القدرة على التعامل مع عدد كبير من أجهزة الحواسيب المصنعة في شـــركات مختلفة Portability .

8. القدرة على التعامل مع شبكات الحاسوب التي تستخدم وحدات طرفية بعيـــدة Remote Terminais .

9. القدرة على إضافة إمكانيات جديدة الى نظام التشغيل أو إلغــــاء إمكانيــات أصبحت غير مطلوبة وذلك دون الحاجة إلى كتابة برنــــامج جديــد لنظــام التشغيل.

(5-3-3) أنواع نظم التشغيل :

يمكن تقسيم نظم التشغيل إلى نوعين أساسيين وعلى الشكل التالي :

أولاً : نظم التشغيل حسب نوع أجهزة التخزين المساعدة والتي يمكن أن تنقسم إلى فرعين هما :

1. نظــام التشـــغيل القرصـــي Disk Operating system واختصاراً DOS ، والقرص على نوعين إما صلب (Hard) أو مرن (Floppy) .

2. نظام التشغيل الشريطي Tape Operating System واختصـــاراً TOS والذي يتم خزن المعلومات فيه على الشريط الممغنط.

ثانياً : نظم التشغيل حسب العلاقة بين المستخدم والحاسوب وعدد المســـتخدمين الذين يعملون في وقت واحد على الحاسوب ، وهذه النظم يمكــــن تقســـيمها إلى أربعة أنواع وهي:

1. نظم تشغبل البرنــــامج الواحـــد Single Program System

في هذا النظام تعمل جميع مكونات الحاسوب على تنفيذ برنامج واحد فقط في الوقت الواحد ، والمستخدم يتعامل بمفرده مع الحاسوب ويراقب بنفسه أجهزة الإدخال والإخراج ، وهذا النوع متوفر في جميع الحواسيب الشخصية .

2. نظـــــام البرمجـــــــة المتعـــــــــددة Multitasking System

وهذا النوع من أنظمة التشغيل يسمح بتخزين اكثر من برنـــــامج واحـــد في الذاكرة الرئيسية وفي نفس الوقت تكون جميع مكونات الحاسوب في حالـــــة تشغيل لعدد من البرامج في آن واحد بحيث يكون أحد البرامج في حالة قـــراءة وآخر في حالة معالجة ، وآخر في حالة طباعة ... والبرامج المـــراد معالجتـــها والتي تنتظر دخولها إلى الحاسوب ، تكون مرتبة بشكل طابور علـــى جـــهاز تخزين مساعد وإن عدد البرامج التي يمكن تخزينها في الذاكرة في وقت واحـــد يعتمد على سعة الذاكرة ونوع نظام التشغيل وحجـــم الـــبرامج المطلـــوب تنفيذها. لذلك تتطلب مثل هذه الأنظمة عدة أمور منها :

1- توفر ذاكرة رئيسية ذات سعة كبيرة .

2- توفر أجهزة تخزين مساعدة سريعة الأداء .

3- توفر وحدة الحساب والمنطق من النوع السريع جدا .

3. نظام المشاركة الزمنية Time Sharing

في هذا النوع من النظم يتم تقسيم وقت المعالجة بين مجموعة من الـــبرامج في الذاكرة الرئيسية بحيث يتم إعطاء الفرصة لجميع البرامج في الحصـــول علـــى نصيب صغير من عمل وحدة المعالجة المركزية بالتساوي و بالتناوب ، ثم ظـــهر فيما بعد نظام التشغيل التفاعلي المتعدد البرمجة ، والذي يسمح لعـــــدد مـــن المستخدمين بالتفاعل مباشرة مع جهاز الحاسوب عن طريق تنفيذ برامجهم مـــن وحدات طرفية متعددة متصلة بالجهاز الرئيسي .

أما أهم ميزات هذا النظام :

- سهولة الاتصال المباشر بين المستخدم وجهاز الحاسوب الرئيسي .
- إتاحة الفرصة لعدد كبير من المستخدمين بالتعامل مع الحاسوب بآن واحد.
- انخفاض التكاليف .

4. نظـــام المشـــاركة الزمنيـــــة الحقيقـــــي Real Time System

وهذا النظام هو تطوير للنظام السابق إذ أن المستخدم لا ينتظر وقتاً معيناً لتنفيـــذ برنامجه بل يكون التجاوب لحظي وفوري مهما يكن عدد المســـتخدمين علـــى الأجهزة الطرفية .

من أمثلتها نظم حجز أماكن الطيران ، نظم المراقبة الجمركيـــة علـــى نقـــاط الحدود..

5. نظام الذاكرة الافتراضية Virtual Memory

يستخدم في الذواكر التي سعتها اكبر من السعة الحقيقية لذاكرة الكمبيوتـــــر ، و يعتمد هذا النظام على تقسيم البرنامج إلى مقاطع وتنفيذ المقطع تلو المقطع .

(5 – 3 – 4) أنواع نظم التشغيل المستخدمة مع الحواسيب الشخصية

يمكن تقسيم أنواع نظم التشغيل المستخدمة مع الحواسيب الشخصية إلى عدة أنواع منها :

1. **نظـــــام التشـــغيل (CP/M-80) Control Program Microprocessors:**

يعتبر هذا النظام نظام التشغيل القياسي للحواسيب الشخصية التي تســـتخدم المعالجات الدقيقة (Z80) و (8080) (المقصود بالمعالجات الدقيقـــة هـــي الميكروبروسيسورز التي تمثل عقل الحاسوب ، وهذه الأنواع مـــن المعالجـــات تستخدم وحدات بيانات طولها (8 bit) .

طور هذا النوع ليصبح صالحاً للاستخدام مع بعض أجـــهزة الأبـــل Apple وأجهزة راديو شاك Radio Shake .

2. **نظام التشغيل (Unix)**

بدأ استخدام هذا النظام عام 1969 على الأجهزة الكبـــيرة و المتوســـطة ثم تم تعديله ليعمل على الحواسيب الشخصية ، و يعتبر هذا النظام من النظـــم ذات

الجودة العالية في الأداء و أقربها إلى نظام التشغيل المثالي الـــذي ســبق شــرح خصائصه بالإضافة إلى إمكانية استخدامه مع الحواسب التي تستخدم وحــدات بيانات (16 bit) و (32 bit) .

يعتمد نظام unix على ثلاث مستويات رئيسية وهي :

- البرنامج القائد Kernel وهو برنامج ينظم المهام ومخازن البيانات .
- برنامج الغلاف Shell وهو برنامج يترجم أوامر المســتخدم حــتى يتمكن الحاسوب من تنفيذها .
- البرامج التطبيقية والمساعدة : Tolls and Applications وهــي مجموعة البرامج المستخدمة مع نظام التشغيل لتضيف له إمكانيــات كثيرة .

3. نظام التشغيل (MS-DOS)

يعتبر هذا النظام اكثر نظم التشغيل انتشاراً بسبب استخدامه كنظــام تشــغيل لأول جهاز حاسوبي شخصي تنتجه شركة IBM ، وقد تم تعديله عدة مــرات وما زال لحد الآن تظهر نماذج معدلة منه وقد بدأ بأول إصدار من البرنامج وهــو (DOS 1.0) ثم استمرت الشركة المنتجة في إنتاج إصدارات متعــددة حــتى وصل إلى الإصدار (DOS 6,22) .

ويتميز هذا النظام بالإضافة إلى خصائص نظام التشغيل المثالي المشروحة سـلبقا ، أنه سهل الاستخدام ويعمل على الأجهزة التي تستخدم المعالج الدقيــق (8088) والتي تعتبر أرخص الأجهزة وتلبي متطلبات الاستخدامات الشخصية والمترليــة للحاسوب .

4. نظام التشغيل (MS-OS/2)

يعتبر هذا النظام اكثر تطوراً من نظام MS-DOS ومستخدم مع الأجــهزة (PS/2) المنتجة بواسطة شركة IBM له ميزات متعددة منها استخدام النوافــذ

في اختبار الأوامر المطلوبة وذلك بالإضافة إلى استخدام الطريقـــة المعتـــادة في إدخال الأوامر عن طريق سطر الأوامر بالإضافة إلى امتيازه بخصـــائص نظـــام التشغيل المثالي ما عدا إمكانية الاستخدام المتعدد .

ورغم ميزاته فانه ليس منتشراً أو شائع الاستخدام مثل نظـــام MS – DOS لكونه يعمل على الأجهزة التي تستخدم المعالج الدقيـق (80286) و (80386) وهي اكثر تكلفة من المعالج (8088) .

(5 – 4) نظام التشغيل MS-DOS

كما سبق الإيضاح فإن هناك العديد من نظـم التشـغيل الـتي تختلـف في خصائصها وإمكانياتها ، وقد تم اختيار نظـــام التشـغيل DOS لدراســته في هذا المقرر لشيوعه وانتشاره بالإضافة إلى ســـهولة اســتخدامه وبســاطته ، ولا يمكن لمســتخدم الحاسـوب الشـخصي الاســتغناء عنــه حيــث أن الــبرامج التطبيقية التي تعمل معه كثيرة ومتعـــددة .

نظام التشغيل DOS هو النظام الذي يقود عمل الحاســـوب ويشــكل واســطة ربط بين المستخدم الحاسوب و(MS-DOS) وهـــذه تمثــل الحـــروف الأولى من الكلمــات: Microsoft Disk Operating System وتعــني نظـــام التعامل مع الأقراص ، وهذا النظام من إنتـــاج شــركة مايكروســوفت ، وقـــد بدأ العمل بهذا النظام علـــى 1981 في شــركة مايكروســوفت Microsoft أو IBM لتشــغيل أجــهزة الحواســيب الشـــخصية المنتجــة في شـــــركة IBMوقد تم تعديله عدة مرات وما زال يتم تعديلــــه حـــتى الآن كمـــا ذكرنـــا ، وقد بدا بأول إصدار مـــن البرنـــامج باســـم (MS-DOS Version 1) أي نظام التعامل مع الأقراص – الإصدار (1) ثم اســـتمرت الشــركة في إنتـــاج

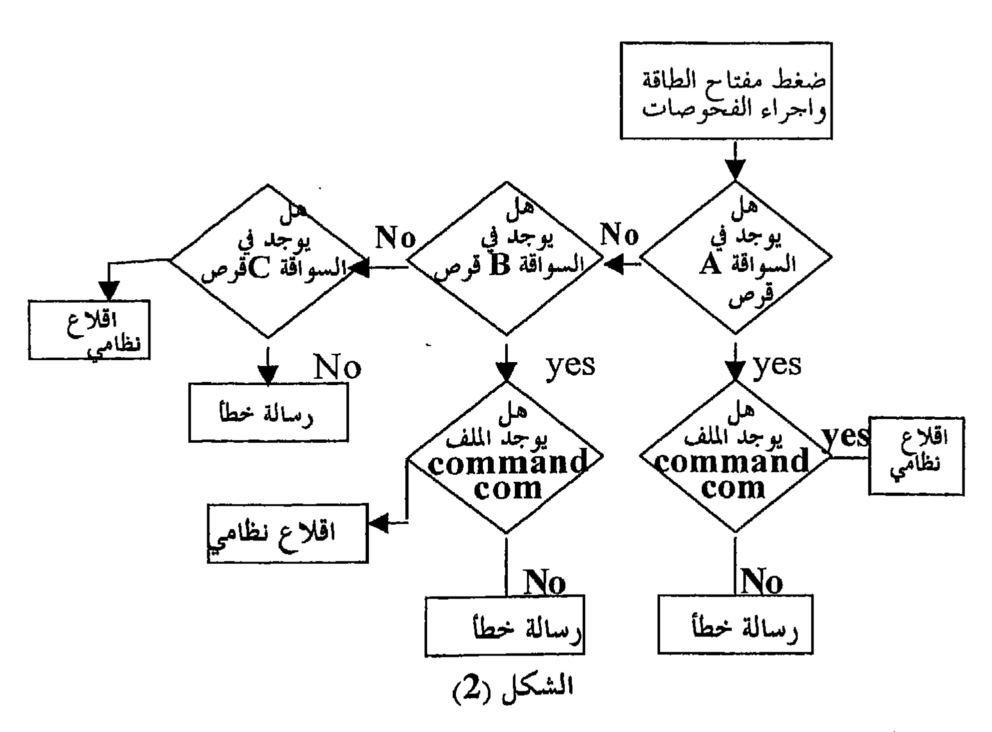
وتطوير هذا النظـــام حـــتى وصلـــت إلى الإصـــدار 6.22 (MS-DOS 6.2) وهو نظام التشغيل المستخدم في جميع أجهزة الحاســـوب الشـــخصية.

ونشير هنا إلى أن تطوير نظـــام التشـــغيل MS-DOS ارتبـــط بتطويـــر المعالجات الدقيقة (Microprocessors) حيث أن معظـــم الحواســـيب الـــتي تستخدم نظام التشغيل MS-DOS تحتوي على عائلـــة المعالجـــات الدقيقـــة ، والـــتي تشـــتمل المعالجـــــات (8086) (8088) (80286) (80386) (80486) .

(هذه المعالجات هي عبارة عـــن شـــرائح صغـــيرة (Chips) تحتـــوي علـــى آلاف من الدوائر الإلكترونيـــة المتكاملـــة (Integrated Circuits) الـــتي تنتج بواسطة شـــركة إنتـــل (Intel Corporation)) .

(5–4–1) تشغيل الحاســـوب

بعد الضغط على مفتـــاح الطاقـــة لتشـــغيل جـــهاز الحاســـوب تقـــوم الروتينات الموجـــودة في ذاكـــرة ROM بـــإجراء فحوصـــات عامـــة علـــى معلومات الحاسوب حيـــث يتـــم قـــراءة الســـواقة A أولاً ثم B ثانيـــاً ثم C ثالثاً، وبعد إنهاء هذه الفحوصات يتم الإقلاع مـــن إحـــدى الســـواقاتA أو B أو C ، وفي كل الحالات يجب أن يتواجـــد علـــى أحـــد الســـوقات ملـــف الإقلاع (Command.com) والشكل (2) يمثل إقـــلاع الحاســـوب .



الشكل (2)

ملاحظة (1) :

في حالة وجود ملف الإقلاع على السواقة C فإنه يحذر من وضع الأقراص في السواقات الأخرى إلا بعد ظهور الرسالة التالية : Starting MS-DOS .

ملاحظة (2):

كلمة ملف هي عبارة عن مجموعة المعلومات المرتبطة مع بعضها والتي يمكن تخزينها في الحاسوب (على أحد الأقراص) ، وأحد أهم ملفات النظام هو ملف الأوامر الذي يسمى عادة بـ Command.com و يضم الأوامر (الروتينات) الأساسية لنظام التشغيل .

ملاحظة (3) :

إذا تم الإقلاع من السواقة C عندئذ يظهر في الزاوية اليسرى العليا من الشاشة :

C : \>

والذي يدعى محث النظام System Prompot أو محث الأوامرCommand

Prompot أو مؤشر الدوس .

وإذا تم الإقلاع من السواقة A (و الذي يكون فيها قرص مرن) عندئـــــذ سيظهر المحث بالشكل التالي : A:\>

و أمام محث النظام يظهر خط صغير ينبض بشكل مستمر (–) يدعـــى المشـــيرة Cursor وهي تشير مكان البدء في الكتابة على الشاشة .

(5 – 4 – 2) أوامر نظام التشغيل MS-DOS

إن نظام التشغيل MS-DOS ما هو إلا عبارة عن مجموعة مـــن الأوامـــر والتعليمات التي تمكن المستخدم من التعامل مع الحاسوب مثل إدارة الملفات مــــن نسخ وحذف وكتابة وتسمية ملف ، أو التعامل مع الأقراص و تهيئتها أو التعـــامل مع المكونات الأخرى للحاسوب .

تقسم أوامر نظام التشغيل MS-DOS إلى مجموعتين من الأوامر :

- الأوامر الداخلية Internal Command : وهي مجموعة مـــن التعليمـــات الكثيرة الاستعمال والموجودة في ملف واحد هـــو Command . Com ، وهذا الملف ينتقل الى الذاكرة عند تشغيل الحاسوب ، لذلك تنفذ أي من هذه الأوامر من الذاكرة الرئيسية بدون الرجوع إلى القرص ، من هذه الأوامر ,... Copy , Date
- الأوامر الخارجية : External Command هذه الأوامر توجد على هيئـــة ملفات قابلة للتنفيذ ، ولتنفيذ أي أمر لا بد من طلب ملفه من القرص المخـــزن عليه نظام التشغيل لوضعه في الذاكرة المؤقته RAM ومن ثم تنفيذه من هــــذه الأوامر Disktopy , Format ,....

وبشكل عام يمكن تقسم أوامر نظام التشغيل إلى أربعة مجموعات هي :

- أوامر الخدمات .
- أوامر التعامل مع الأقراص .

- أوامر التعامل مع الملفات
- أوامر التعامل مع الفهارس (الأدلة) .

وسنقوم فيما يلي بشرح أهم الأوامر المتعلقة بكل مجموعة بما يلبي حاجة الطالب في هذا المقرر للبدء في العمل على الحاسوب .

(5-4-2-1) أوامر الخدمات :

- **الأمر VER**

وظيفته : للتعرف على رقم الإصدار للنظام المعمول به على الحاسوب

صيغته :

C:\> VER ↵

الشرح : إن المقصود بالإشارة ↵ هو ضغط مفتاح Enter الإدخال ، ونتيجة تنفيذ هذا الأمر تظهر رسالة تبين رقم الإصدار للنظام المعمول به كما يلي:

MS-DOS Version 6.2

- **الأمر Date**

وظيفته : عرض التاريخ الحالي وإمكانية تصحيحه إذا لزم الأمر .

صيغته :

C:\> DATE ↵

الشرح : تظهر رسالة تعرض التاريخ الحالي مع إمكانية تصحيحه إذا كان خطأ كما هو موضح في الرسالة (mm-dd-yy) أي ندخل الشهر– اليوم – السنة ثم نضغط مفتاح الإدخال Enter ، وإذا لم يكن خطأ فإننا نضغط مفتاح الإدخال للرجوع إلي محث النظام .

- **الأمر Time**

وظيفته : عرض الوقت الحالي وإمكانيه تصحيحه إذا لزم الأمر .

صيغته :

C:\> TIME ↵

الشرح : تظهر رسالة تعرض الوقت الحالي فإذا كان صحيحاً نضغط مفتاح الإدخال للرجوع إلي محث النظام وإذا كأن خطأ فأنه يمكن تصحيحه كما هو موضح في الرسالة : (hours:minutes) (a/p) أي الساعة ثم الدقائق ثم a للإشارة إلى التوقيت صباحي أو p للإشارة إلى التوقيت المسائي ثم نضغط مفتاح الإدخال Enter للرجوع الى محث النظام .

- **الأمر CLS :**

وظيفته : تنظيف الشاشة .

صيغته :

C:\> CLS ↵

الشرح : يؤدي إلى أنتقال محث النظام والمشيرة إلي الزاوية اليسرى العليا من الشاشة وتصبح الشاشة نظيفة من أية كتابات .

- **الأمر Doskey :**

وظيفته : جعل التعامل مع مؤشر DOS (محث النظام) أسهل من ذي قبل وله فوائد كثيرة :

1. يتيح استخدام مفاتيح الأسهم للانتقال الى الحرف المخطوء ثم تصحيحه فقط دون الحاجة الى مسح الحروف اللاحقة للحرف المخطوء .
2. يتيح استرجاع الأوامر التي سبق أن أدخلت بعد تشغيل الجهاز عند مؤشر الدوس ، لذلك لا حاجة لإعادة كتابة الأوامر التي سبق إدخالها عند الحاجة أليها مرة أخرى أثناء العمل ، ولكن نستخدم مفاتيح الأسهم العلوي-السفلي لاستعراض الأوامر المدخلة إلي أن نصل إلى الأمر المطلوب إعادة كتابته :

صيغته :

C:\> DOSKEY ↵

الشرح : لا شيء يحدث عند إستخدام الأمر السابق إلا أنه بواسطته يتم تعريف المفاتيح اللازمة للتحرير والاسترجاع التالية :

مفاتيح التحرير :

اسم المفتاح	وظيفته
→	الانتقال حرف واحد في كل مرة إلى اليمين
←	الانتقال حرف واحد في كل مرة إلى اليسار
Ctrl + →	الانتقال كلمة واحد في كل مرة إلي اليمين
Ctrl + ←	الانتقال كلمة واحد في كل مرة إلي اليسار
Home	الانتقال إلى بداية الأمر
End	الانتقال إلى نهاية الأمر
Esc	مسح الأمر (حذف السطر الذي كتب منذ لحظات)
Del	مسح الحرف الذي تقع عليه المشيرة
Backspace	مسح الحرف الذي يقع على يسار المشيرة
Ins	حشر الأحرف ضمن الكلمات .

مفاتيح الاسترجاع :

اسم المفتاح	وظيفته
	استرجاع أوامر الدوس الذي سبق إدخالها قبل الأمر الظاهر عند مؤشر الدوس.
	استرجاع أوامر الدوس الذي سبق إدخاله بعد الأمر الظاهر عند مؤشر الدوس
PgUp	استرجاع أول أمر دوس تم إدخاله
PgDn	استرجاع آخر أمر دوس تم إدخاله
F7	استرجاع جميع الأوامر التي سبق إدخالها مرقمة بشكل متسلسل حسب إدخالها
F9	عند الضغط عليه تظهر الرسالة : Line Number نكتب بعدها رقم أحد الأوامر التي ظهرت نتيجة استخدام المفتاح F7 ثم نضغط مفتاح الإدخال ليصار الى تنفيذ الأمر مباشرة

بعض المفاتيح الضرورية :

اسم المفتاح	وظيفته
أحد الأحرف + Shift	استخدام الأحرف والرموز المكتوبة أعلى كل مفتاح .
Ctrl + c	إنهاء عمل أي أمر
Ctrl +Z أو(Pause)	إيقاف تنفيذ أي أمر وللمتابعة نضغط أي مفتاح من لوحة المفاتيح

(5-4-2-2) أوامر التعامل مع الأقراص :

- **الأمـــر Format :**

وظيفته :

1. تشكيل (تهيئة) القرص المرن أو الصلب لكي يكون جاهزاً لخزن المعلومات.
2. فحص القرص بحثا عن المسارات التالفة و تحديدها لتلافي الخزن عليها .

صيغته :

C:\> FORMAT d : ↵

حيث d تشير إلى اسم السواقة التي تحوي القرص المراد تهيئته ، أي يمكن أن يكون a أو b أو c .

الشرح : يقوم الدوس بعمليـــة التهيئـــة المكونـــة مـــن ثلاثـــة خطـــوات :

1. تقســيم القـــرص الى مســـارات (Tracks) وقطاعـــــات (Sectors)

المسار : هو عبارة عن دائرة على القرص مركزها هـــو مركـــز القـــرص .

القطاع : هو المنطقة المشكلة بين مسارين ونصفي قطرين .

2. إنشـــاء جـــدول توزيـــع الملفـــات (File Allocation Table) :

جدول توزيع الملفات : هو جدول الذي سيحوي أسماء الملفات المخزنة على القرص وعناوين تخزينها على القطاعات .

3. إنشـــاء دليـــل الجـــذر للقـــرص (Root Directory)

وعند الانتهاء من عملية التهيئة تظهر رسالة يطلب فيها إدخال اسم القـــرص إذا أردنــــا أو نضغـــط مفتـــاح الإدخـــال Enter

ثم يتم عرض المعلومات المتعلقة بالقرص المهيأ وسؤال إذا كان لدينا قرص آخـــو لتهيئته إما نكتب Y ثم Enter أو N ثم Enter .

ملاحظات :

1. اسم القرص يجب أن يكون مؤلف من 11 حرفا كحـــد أقصـــى ويمكـــن أن يتضمن الفراغ ، غير أنه ليس بالإمكان استخدام الرموز التالية في اسم القرص:

\ ? * / : , ; = + & ^ ( ) [ ] < " >

2. إذا قمنا بتهيئة قرص يتضمن بيانات مخزنة مسبقاً، لذلك يجب الانتباه الى خلـــو القرص أو أننا لسنا بحاجة للبيانات الموجودة عليه .

3. إن وجود القطاعات التالفة في القرص لا تجعله غير صالحـــاً بـــل أن الأمـــر Format سيقوم بتحديدها حتى لا يقوم الدوس بخزن المعلومات عليها .

- الأمـــر **Unformat :**

وظيفته : إلغاء تهيئة الأقراص وإعادة القرص وملفاته إلى الحالة التي كانت عليها قبل عملية التهيئة .

صيغته :

C:\> UNFORMAT d : ↵

الشرح : إعادة القرص إلي وضعه قبل التهيئة .

ملاحظات :

غالباً ما يستخدم هذا الأمر عندما نريد استرجاع الملفات والفهارس التي تم مسحها عند تهيئة القرص بدون قصد ، وهذا يتم فقط إذا لم نقم بتخزين بيانـــات جديدة على القرص بعد. ومع ذلك يجب أن نعلم أن استخدام الأمر Unformat قد يؤدي إلى فقدان تلك البيانات أو بعضها لذلك يجب اخذ الحيطة عند اســتخدام

الأمر Format ، وأن نفترض دوما أن استخدامه سيؤدي بصورة دائمة إلي فقـدان البيانات الموجودة على القرص .

- **الأمـــــر VOL :**

وظيفته : إظهار أسم القرص .

صيغته :

C:\> VOL   d : ↵

الشرح : يعرض رسالة يوضح فيها اسم القرص إذا كنا قد سميناه مسبقا إمـا عند التهيئة أو باستخدام الأمر التالي .

- **الأمـــــر Label :**

وظيفته : - إعادة تسمية القرص باسم جديد دون الحاجة إلى إعادة تشكيله .

- تسمية القرص إذا لم يكن له اسم .

صيغته :

C:\> LABEL   d : ↵

الشرح : يعرض الدوس رسالة يبين فيها اسم القرص (إذا كان له اسم) ويطالب إدخال اسم جديد (إذا أردنا) ثم نضغط مفتاح Enter .

ملاحظة : الانتقال إلى القرص الموجود على إحدى السواقات نستخدم الصيغة التالية:

الصيغة :

dd:\> d : ↵

حيث dd السواقة الحالية ، d السواقة التي ستنتقل أليها .

مثال : أنتقل إلى السواقة A بعد وضع قرص مرن فيها مع العلم أن السواقة الحالية هي C :

C:\> a : ↵

والنتيجة يصبح محث النظام بالشكل A:\> والذي يمكن استخدام الأوامر عنـــــده أيضاً .

(5-4-2-3) أوامر التعامل مع الملفات :

عندما نقوم بخزن المعلومات على الأقراص (المرنــــة أو الصلبـــة) فـــإن الحاسوب يقوم بحفظ المعلومات على هيئة ملف منفصل (تشبه هذه العملية تمامـــا عملية وضع كافة أوراق أحد التقارير في ملف واحد) وقبل أن يحفظ المعلومـــات يطلب منا تحديد اسم لذلك الملف كي يستطيع استرجاعه عندما نطلب منه .

يستخدم الحاسوب عادة نوعان من الملفات :

1. ملفات نصية Text File : تحتوي على محارف ورموز يمكن قراءتها .
2. ملفات أوامر Command File: وتحتوي على مجموعة من الأوامر الـــتي يحتاجها النظام DOS لتنفيذ أوامره ومحتواها غير قابلة للقراءة لأنها مكتوبـــة بلغة الآلة .

يتكون اسم الملف من جزئين رئيسيين يفصل بينها نقطة (.) dot .

الجزء الأول : اسم الملف file name ويتألف من ثمانية محارف على الأكثر .

الجزء الثاني : الامتداد Extension ويتألف من ثلاثة محارف على الأكثر .

ويمكن استخدام كافة الحروف (الصغيرة و الكبيرة) والأرقام والرموز باســـتثناء الرموز التالية :

? ; ! > < *+ :] [\ / . الفراغ

وللامتداد فائدة كبيرة للحاسوب إذ يساعد على تمييز نوع الملف ، إذ أن الحاسوب يتعامل عادة مع امتدادات نموذجية نذكر منها :

الامتداد	يعبر عن ملف
LTR	رسالة
TXT	نص
DWG	برنامج رسم
DOC	وثيقة Document
COM	برنامج أوامر
EXE	برنامج قابل للتنفيذ
SYS	برنامج تابع للنظام
HLP	برامج مساعد
BAS	برنامج مكتوب بلغة البيسك
PAS	برنامج مكتوب بلغة باسكال
FOR	برنامج مكتوب بلغة الفورتران
BAT	برنامج تنفيذي تدفقي

- **الأمـــــر DIR :**

وظيفته :استعراض الملفات الموجودة على القرص (المرن أو الصلب) .

صيغته :

1) C:\> DIR : ↵

2) C:\> DIR d : ↵

الشرح : إن الصيغة الأولى تؤدي إلى استعراض جميع الملفات الموجودة علـــى القرص الصلب و الصيغة الثانية تؤدي إلى استعراض جميع الملفـــات الموجودة على القرص المرن الموجود في السواقة d

(حيث d اما A اوB) والعودة إلي محث النظام C:\> .

إن الاستعراض يتضمن أسماء الملفات والأمتدادات – سعة كـــل ملـــف – تاريخ ووقت إنشاء الملف أو آخر تعديل على الملف في نهاية الاستعراض يعـرض عدد الملفات وسعتها الإجمالية والسعة المتعلقة على القرص .

ملاحظات :

1. غالباً ما يكون عدد الملفات كبيراً ولا يتسع على الشاشة وفي هذه الحالة الأمر DIR سيعرض آخر مجموعة ملفات (أي الصفحة الأخيرة ، حيث اعتبرنا أن سعة الشاشة تمثل صفحة ، وبالتالي لاستعراض الملفات صفحة صفحة نستخدم الصيغة .

C:\> DIR /P ↵

C:\> DIR /P d: ↵

2. لاستعراض فقط جميع أسماء الملفات بدون السعة والتاريخ والوقت وعلى صفحة واحدة نستخدم الصيغة :

C:\> DIR /W ↵

C:\> DIR /W d: ↵

3. لاستعراض جميع الملفات ذات صفة مشتركة نستخدم :

إما (*) كبديل عن مجموعة أحرف قد تصل إلى ثمانية أحرف .

أو (؟) كبديل عن حرف واحد .

أمثله : - استعراض جميع الملفات على السواقة C ذات الامتداد TXT :

C:\> DIR *.TXT ↵

- استعراض جميع الملفات على السواقة A ذات الاسم Command:

C:\> DIR A: Command.* ↵

- استعراض جميع الملفات على السواقة C التي يكون الحرف الأول من الاسم مجهول وبقية الاسم TEL .

C:\> DIR ?TLE . * ↵

- **الأمر Copy :**

وظيفة : 1) إنشاء ملف نصي

صيغته :

C:\> copy con اسم الملف . الامتداد ↵

الشرح : الأمر السابق مفاده الطلب من نظام الدوس نسخ ما يجري إدخاله من لوحة المفاتيح ووضعها تحت اسم الملف المعطى .

وتنفيذ الأمر السابق يؤدي إلى ظهور المشيرة على السطر التالي ، نبدأ مباشرة بكتابة أي نص باللغة الإنجليزية ، ولإنهاء الكتابة والتخزين نستخدم المفتاحينCtrl + Z ، ثم مفتاح Enter.

ملاحظة :

1. يعتبر هذا الأمر أبسط محرر نصوص يعمل تحت نظام التشغيل MS-DOS ولكن مساوئه أنه لا يمكننا من العودة الى السطر السابق للكتابة أو التصحيح بعد الضغط على مفتاح الإدخال ENTER بالإضافة الى أننا لا يمكننا إضافة أو حذف أي حرف بعد تخزينه .
2. إذا أردنا إنشاء ملف نصي على قرص مرن موجود في إحدى السواقات نضيف فقط قبل اسم الملف اسم السواقة ونقطتين رأسيتين كما يلي :

C:\> Copy Con a: إسم الملف . الإمتداد

وظيفة : 2) نسخ ملف من سواقة إلى سواقة أخرى أو إلى نفس السواقة مع تغيير اسم الملف صيغته العامة :

صيغته :

C:\> Copy d1: اسم الملف . الامتداد d2: اسم الملف الجديد . الإمتداد

الشرح : يقوم النظام DOS نسخ محتوى الملف الموجود على القرص الموجود في السواقة (C,B,A)d1 إلى القرص الموجود في السواقة (C,B,A)d2 تحت اسم جديد .

ويمكن استخدام اسم الملف الأصلي إذ كنا ننسخه إلى قرص موجود في سواقة أخرى .

أمثلة :

1 - C:\>COPY   C: MM.TXT   a: ↵
2 - C:\>COPY   a:MM.XTX   a: MAX.TXT ↵
3 - C:\>COPY   *.TXT   a: *.BAK ↵
4 - C:\>COPY   a: Come.doc   C: ↵

وظيفة : 3) جميع مغلقين أو اكثر في ملف واحد

صيغته :

↵الامتداد.اسم الملف الجديد الامتداد.اسم الملف(2) + الامتداد . اسم الملف(1)
C:\>COPY

الشرح : يقوم الـ DOS بجمع الملفين (1) و (2) في ملف تحت اسم الملف الجديد.

ملاحظات :

1) إذا كانت جميع الملفات في أماكن مختلفة ندخل قبل اسم الملف السواقة التي تحوي القرص الذي يحوي الملف المطلوب مع نقطتين رأسيتين (:) كما يلي :

C:\>Copy   a: Game.TXT + C:Prog.TXT   a:MM.doc ↵

2) إذا لم نكتب اسم الملف الجديد فإن الملف الناتج عن الجمع يأخذ اسم الملف الأول وامتداده .

3) يمكن جمع اكثر من ملفين بنفس الأسلوب .

- **الأمر Move :**

وظيفته : نقل ملف .

صيغته :

C:\> Move d1: اسم الملف . الامتداد d2: أسم الملف . الامتداد

الشرح : يقوم الدوس بحذف الملف الأصلي من القرص الموجود في السواقة d1 وتخزينه على القرص الموجود في السواقة d2 إما باسم الملف الأصلي حسب ما ندخل .

مثال :

C:\>Move a: MM.TXT C: ↵
C\:>Move a:MM.TXT C: NCX.TXT ↵

• الأمـــــر DEL :

وظيفته : حذف الملف .

صيغته : ↵ الامتداد. اسم الملف C:\> DEL d:

الشرح : يقوم الدوس بحذف الملف الموجود على القرص الموجـــود في الســـواقة (C,B,A)

ملاحظات :

1- إذا لم نضع أمام اسم الملف اسم السواقة يقوم الدوس بالتفتيش عن الملـــف المطلوب في السواقة C ويحذفه.

2- يمكن إجراء حذف مجموعة ملفات باستخدام * أو ؟

C:\>DEL a:*.TXT ↵

3- عملية الحذف تؤدي الى حذف الملف نهائياً ويمكن استرجاعه أحيانـاً إذا لم نقوم بأي عملية على القرص الذي يحوي الملف المحذوف وذلـــك باســتخدام الأمر التالي :

• الأمـــــر Undelete

وظيفته : إلغاء حذف الملف شريطة أن لا نكون قد قمنا بتخزين أو نسخ على القرص الذي كأن يحوي الملف المحذوف .

صيغته :

↵ الامتداد اسم الملف C:\>UNDELETE D:

الشرح : سيعرض الدوس رسالة يظهر فيها اسم الملف ما عدا الحرف الأول ثم يعرض رسالة يسأل فيها إذا كنا نريد فعلاً إلغاء عملية الحذف أم لا ، فإذا كنـــا

نريد فعلاً إلغاء عملية الحذف واسترجاع الملف المحذوف نضغط على الحــــرف Y (أيYes) ثم نكتب الحرف الأول من اسم الملف ثم نضغط على مفتاح ENTER فيعرض الدوس سؤال آخر إذا كنا نريد إلغاء حذف ملف آخر وعندئـــذ نضغـــط على Y ونكتب الحرف الأول من اسمه .... أو نضغط N .

ملاحظة هامة : عندما نقوم بحذف الملفات فإن الدوس في الواقع لا يقـــوم بحذفـــه نهائيا بل يقوم بحذف الحرف الأول من اسمه وبذلك يهيئ المساحة التي كان يشغلها الملف المحذوف علي القرص لخزن أي ملف لاحق نقوم بخزنه ، وما لم نقوم بخــــزن أي ملف على القرص بعد إجراء عملية حذف الملفات فإنه لدينا فرصـــــة لإلغـــاء عملية الحذف .

- **الأمــــــر Type :**

وظيفته : استعراض محتوى ملف معين .

صيغته :

C:\>Type d: اسم الملف. الامتداد ↵

الشرح : استعرض محتوى الملف الموجود على القرص الموجود على الســــــواقة (A أو B أو C) .

وإذا لم نكتب اسم السواقة فأنه يقوم الدوس بالتفتيش عـــن الملـــف المكتـــوب في السواقة C لاستعراضه .

ملاحظات :

1. الملفات ذات اللاحقة EXE أو COM ستكون مكتوبة بلغة الآلة وبالتـــــالي عند استعراض تلك الملفات تظهر شفرة غير مقروءة .
2. إذا كان سعة الملف اكبر من سعة الشاشة عندئذ يمكن استخدام الأمر التـــــالي لاستعراض جزءاً جزءاً من الملف .

C:\>TYPE d: اسم الملف الامتداد | MORE ↵

حيث | أحد مفاتيح لوحة المفاتيح .

• **الأمـــــر Print :**

وظيفته : طباعة محتوى ملف على الطابعة .

صيغته :

C:\>PRINT d: اسم الملف . الامتداد ↲

الشرح : يتم نقل محتوى الملف المطلوب إلى الطابعة ليصار إلى طباعته علـــى الورق.

• **الأمـــــر Ren :**

وظيفته : إعادة تسمية الملف .

صيغته :

C:\>REN d: اسم الملف.الامتداد الاسم الجديد للملف.الامتداد ↲

الشرح : يقوم الدوس بتغيير الاسم القديم للملف الموجود علـــى القـــرص الموجود على السواقة d (حيث d يمكن أن يكون A,B,C) الى الاسم الجديد.

وإذا كان الملف على السواقة C فلا حاجة لكتابة (: C) قبل اسم الملف .

ملاحظة : يمكن إعادة تسمية مجموعة من الملفات وذلك باستخدام * أو ؟

مثال : C:\>REN *.TXT *. DOC:↲

• **الأمـــــر Attrib :**

وظيفته : إظهار الخواص المميزة للملف مع إمكانية التعديل في هذه الخـــواص (إضافة أو حذف) .

صيغته :

1. التعرف على شكل الملف

C:\> Attrib d: اسم الملف . الامتداد ↲

2. تغيير الصفة :

C:\> Attrib (+ أو -) الصفة اسم الملف . الامتداد ↵

الشرح : أي ملف يمتلك صفات معينة ترشد الحاسوب ونظام التشغيل إلى كيفية التعامل مع هذا الملف وهذه الصفات هي :

رمز الصفة	اسم الصفة	وظيفتها عندما تكون مصاحبة للملف
R	Read Only قراءة فقط	لا يمكن تعديل الملف أو الكتابة أو حذفه ولكن فقط القراءة
H	Hidden الإخفاء	إخفاء الملف من قائمة الملفات عند استخدام أمر Dir .
S	System نظامي	الملف مخصص لاستخدامات النظام .
A	Archive الأرشفة	تشير إلى أن الملف جديد أو أنه قديم ويحتاج في كلا الحالتين إلى نسخة احتياطية .

إن استخدام صيغة التعرف على الملف تؤدي إلي ظهور الرسالة ذات الشكل التالي:

اسم السواقة \ اسم الملف . الامتداد

أما استخدام الصيغة الثانية فأن إشارة + تشير إلى إضافة الصفة ، و إشارة(-) تشير إلى حذف الصفة.

ملاحظات :

1) يمكن استخدام الصيغتين السابقتين من اجل مجموعة من الملفات باستخدام * أو؟

2) يمكن إجراء عملية الحذف والإضافة لأكثر من صفة في أن واحد .

(5-4-2-4) أوامر التعامل مع الفهارس :

لنستعرض المثال التالي :

بفرض أنه لدينا في قسم الأرشيف لشركة ما الملفات التالية:

1-ملفات خاصة بالموظفين وتضم :ملفات خاصة بالحضور والغياب .

ملفات خاصة بالرواتب والأجور والتعويضات.

ملفات خاصة بالعقوبات .

2-ملفات خاصة بالإدارة وتضم :ملفات خاصة بالتعليمات .

ملفات خاصة بالشؤون الإدارية.

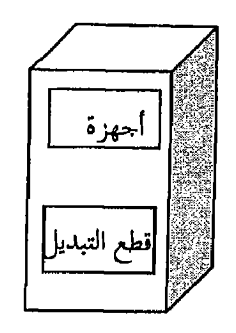
3-ملفات خاصة بالمعدات وتضم :ملفات خاصة بالأجهزة .

ملفات خاصة بقطع التبديل .

عندئذ أمام موظف الأرشيف أحد الخيارين :

الخيار الأول : وضع الملفات فوق بعضها البعض دون ترتيب ، وفي هـذه الحالـة عندما يطلب منه إحضار ملف معين ، فإنه يضطر إلى البحث في كـل الملفات كي يجد طلبه لـه، وهذا ربما يستهلك وقتاً وجهداً كبيرين .

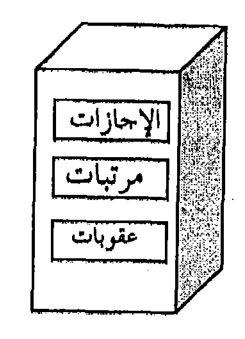
الخيار الثاني : ترتيب الملفات وتنظيمها ضمن خزائن وأدراج كما في الشكل (3).



ملفات خاصة بالمعدات



ملفات خاصة بالإدارة



ملفات خاصة بالموظفين

الشكل (3)

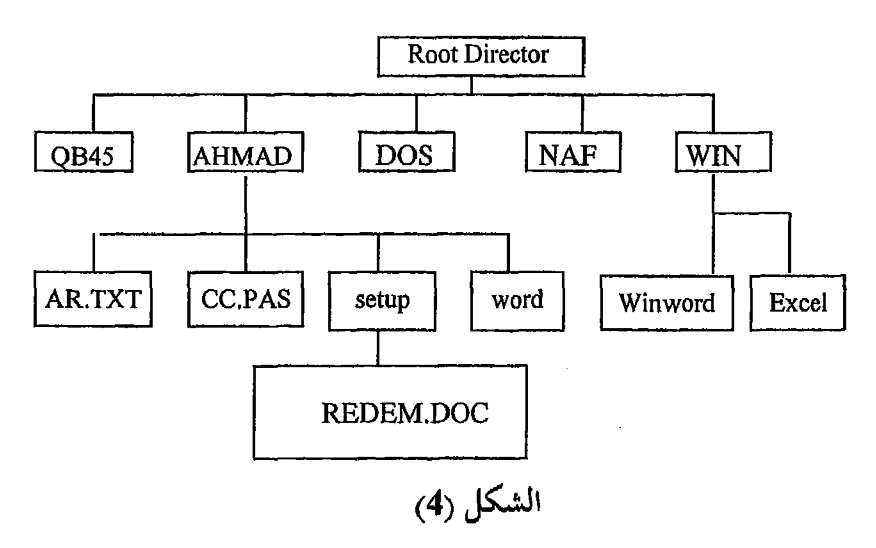
وفي هذه الحالة عندما يطلب منه إحضار ملف معين تصبح العملية سهلة حيث يتوجه مباشرة إلى الخزانة التي تحوي هذا النوع من الملفات ون ثم إلى الدرج المناسب وبذلك يختصر وقت وجهد كبيرين .

إن الخيار الثاني في تنظيم غرفة أرشيف الشركة يشبه تماما عملية تخزين الملفات وتنظيمها بواسطة نظام التشغيل سواء كان التخزين على القرص المرن أو القرص الصلب ولكن باستخدام الفهارس (أو الأدلة) (Directories) والفهارس (الأدلة) الفرعية Subdirectories التي يقوم نظام التشغيل MS-DOS بإنشائها باستخدام مجموعة من الأوامر .

يمثل القرص (الصلب أو المرن) الفهرس الجذر (الرئيسي) Root Directory الذي يحتوي على ملفات وفهارس فرعيه ، والفهارس الفرعية هي عبارة عن منطقة تخزينية من القرص لها اسم معين ويمكن أن تحوي على ملفات وفهارس فرعية أيضا ، وعندما نقوم بتهيئة أي قرص جديد باستخدام الأمر Format فإن نظام الدوس يقوم بإنشاء فهرس الجذر ويعتبر هذا الفهرس هو فهرس افتراضي لحين الانتقال إلى فهرس آخر ، ونظام الدوس يخصص رمز الشوطة المائلة \ للفهرس الجذر ولا يسمح نظام الدوس بحذف الفهرس الجذر ، وأما الفهرس الفرعي فهو أي فهرس باستثناء الفهرس الجذر.

اسم الفهرس يتألف من ثمانية محارف كما في تسمية الملفات باستثناء الامتداد وعند استعراض الملفات بالأمر DIR فأنه يتم عرض بالإضافة الى الملفات أسماء الفهارس الفرعية الموجودة ضمن الفهرس الجذر وأمام الأسم مكتوب <DIR> .

والشكل (4) يبين التركيب الهيكلي للفهارس والملفات في قرص على شكل شجرة مقلوبة (الشجرة افتراضية) :



الشكل (4)

للوصول الى الملف REDEM.DOC الموجود في الفـــهرس الفرعـــي SETUP المتفرع عن الفهرس الفرعي AHMAD المتفرع عن الجذر في القرص الموجـــود في السواقة A نكتب عند مؤشر الدوس ما يلي :

C:\>{احد اوامر الدوس}A:\AHMAD\SETUP \REDEM.DOC

السواقة الحالية – اسم السواقة الموجودة فيها ا لقرص – علامة الفهرس الجذر – فهرس فرعي – فاصلة ما بين الفهارس – فهرس فرعي – فهرس فرعي – فاصلة ما بين الفهارس – الملف المطلوب

إن سلسلة الفهارس والفهارس الفرعية التي يتوجب على نظام الـــدوس ســـلوكها للوصول الى الملف المطلوب يدعى المسار (PATH).

وعند الضغط على مفتاح الإدخال ENTER للشكل الســـابق فســـوف يقـــوم الدوس بالبحث في القرص الموجود في السواقةA وذلك ابتداء من الفهرس الجذر \ حيث يبحث عن الفهرس الفرعي AHMAD ومن ثم يبحث ضمـــن الفـــهرس الفرعي AHMAD عن الفهرس الفرعي SETUP وبعدئـــذ يبحـــث ضمـــن الفهرس الفرعي SETUP عن الملف المطلوب REDEM.DOC ، بعد ذلـــك يطبق على الملف المطلوب أمر الدوس المعطى عند محث النظام مباشرة (أمر الـدوس أي أمر من أوامر الدوس التي مرت معنا سابقاً أو التي ستمر لاحقاً) .

ملاحظة (1) : يوجد طريقة ثانية للوصول الى الملف وذلك بالانتقال الى الفهرس الفرعي الذي يحوي الملف المطلوب بالأمر (CD) الذي سنشرحه لاحقا ، ثم نطبق الأمر على الملف ، بمعنى آخر يمكن تنفيذ جميع الأوامر المتعلقة بالنظام دوس ضمن الفهرس الفرعي وكأن الفهرس الفرعي قرص مستقل بحد ذاته .

- **الأمر MD :**

وظيفته : إنشاء فهرس MAKE DIRECTORY :

صيغته : 1) إنشاء فهرس فرعي على القرص C :

C:\>MD أسم فهرس فرعي ↵

2) إنشاء فهرس فرعي على القرص الموجود في إحدى السواقات B,A وسنرمز لها بـ d .

C:\>MD d: أسم فهرس فرعي ↵

3) إنشاء فهرس فرعي موجود ضمن القرص الموجود في أحد السواقات d .

C:\>MD d: المسار \ أسم فهرس فرعي ↵

ويمكن أيضا الانتقال الى السواقة المطلوبة والفهرس الفرعي المطلوب ثم إنشاء فهرس فرعي كما سننشئ ذلك في الأمثلة التابعة للأمر CD.

أمثلة :

1. أنشئ ثلاثة فهارس فرعية في القرص المرن الموجود في السواقة A وهي:

(GAMES,PROGRAM, AHMAD)

C:\>MD A:AHMAD ↵

C:\>MD A:PROGRAM ↵

C:\>MD A:GAMES ↵

2. أنشئ فهرسين ضمن الفهرس الفرعي AHMAD باسم AHMAD1 وAHMAD2

C:\>MD A:AHMAD\AHMAD1 ↵

C:\>MD A:AHMAD\AHMAD1 ↵

3. أنشئ ثلاثة فهارس فرعية ضمن الفــــهرس الفرعـــي AHMAD1 باســـم SS1,SS2,SS3

C:\>MD A:AHMAD\AHMAD1\SS1 ↵

C:\>MD A:AHMAD\AHMAD1\SS2 ↵

C:\>MD A:AHMAD\AHMAD1\SS3 ↵

- **الأمــــر CD :**

وظيفة : الانتقال الى الفهرس فرعي آخر Ching Directory

صيغته :

1. للانتقال الى الفهرس الفرعي الأبن (أي فهرس فرعي داخل فهرس معين)

C:\> CD d: اسم الفهرس الفرعي ↵

C:\> CD d: اسم الفهرس الفرعي ↵

2. للانتقال الى الفهرس الاب (أي الفهرس الفرعي الذي يضم فهرس معين)

C:\> CD .. ↵

3. للانتقال الى الفهرس الجذر مباشرة نكتب :

C:\> CD \ ↵

أمثله : 1. أنشئ فهرسين فرعين ضمن الفهرس Program الموجود على الســـواقة A باسم Program1 و Program2 وذلك بعــــد الانتقـــال إلى الفـــهرس Program مع العلم أن السواقة الحالية C .

الخطوة (1) الانتقال إلى السواقة A وذلك :

C:\> A: ↵

ينتج محث النظام بالشكل A:\>

الخطوة (2) الانتقال إلى الفهرس الفرعي Program وذلك :

A:\> CD   Program   ↵

ينتج محث النظام بالشكل ↵ A:\> Program>

الخطوة (3) إنشاء الفهرسين الفرعين المطلوبين بالتتالي أي :

A:\> Program> MD   Program 1   ↵

A:\> Program> MD   Program 2   ↵

3. أنشئ ثلاثة فهارس فرعية ضمن الفهرس الفرعي Program1 باسم ss3,ss2,ss1 (يترك للطالب) .

• **الأمر Tree :**

وظيفته : عرض شجرة الفهارس للسواقة الحالية .

صيغته :

dd:\> Tree ↵

حيث dd السواقة الحالية المراد عرض شجرتها .

dd:\> Tree d: ↵

حيث d السواقة التي تحوي القرص المراد عرض شجرته .

أمثله : 1. استعرض الفهارس الفرعية على السواقة C :

C:\> Tree ↵

3. استعرض الفهارس الفرعية على السواقة A .

C:\> Tree A: ↵

أو

C:\>A: ↵

A:\> Tree: ↵

ملاحظات :

1- لعرض شجرة الفهارس مع الملفات :

dd:\> Tree   d:\F ↵

حيث dd السواقة الحالية ، d السواقة المراد عرض شجرتها وملفاتها وعندما dd=d لاحاجة لكتابة :d .

2- إذا كان الشجرة لا تتسع في شاشة واحدة يمكن استخدام الأمر التالي للاستعراضها جزءاً جزءاً .

dd:\> Tree   d:\F |More ↵

3- لعرض جميع الملفات و الفهارس الموجودة في الفهرس الحالي الأمر DIR كما مر معنا سابقا .

4- إن جميع الأوامر السابقة التي مرت معنا يمكن استخدامها انطلاقا إما من السواقة الحالية وذلك بتحديد المسار أو بالانتقال الى الفهرس الفرعي المطلوب أينما كان باستخدام CD .

أمثلة :

1. أنشيء ملف نصي في الفهرس الفرعي Ahmad1 باسم MM.TXT

C:\> Coy Con   A:Ahamd\Ahmad1\MM.TXT ↵

ويمكن انشاء الملف بالطريقة التالية :

نتيجة التنفيذ على الشاشة	أوامر الدوس المتتالية
A:\>	1) C:\> A: ↵
A:\> Ahmad>	2) A:\> CD Ahmad: ↵
A:\> Ahmad> Ahmad1>	3) A:\> Ahmad>CD Ahmad1 ↵
ونبدأ بكتابة نص بعد ظهور مؤشر الكتابة على السطر	4)A:\> Ahmad>Ahmad1> copy con MM.TXT ↵

2- انسخ الملف MM.TXT من الفـــهرس الفرعـــي Ahmad1 إلى الفـــهرس الفرعي ss1 الموجود ضمن الفهرس الفرعي Ahmad1 انطلاقاً من السواقة C.

```
C:\>Copy  A:\Ahmad\Ahmad1\MM.TXT
\ss1\MM.TXT   ↵ a:\Ahmad\Ahmad1
```

3-انسخ الملف MM.TXT من الفهرس الفرعي Ahmad إلى الفهرس الفرعـــي SS1 الموجود ضمن الفهرس الفرعي Program انطلاقا من C وباسم جديـــد LS.DOC .

```
C:\>Copy  A:\Ahmad\Ahmad1\MM.TXT
a:Ahmad\Ahmad1 \SS1\LS.DOC  ↵
```

- **الأمـــــر RD :**

وظيفته : حذف أحد الفهارس الفرعية .

صيغته :

dd:\> RD اسم الفهرس المراد حذفه ↵

إلا أنه لا يمكن تطبيق هذا الأمر مباشرة قبل حذف جميع الملفات و الفـــــهارس الفرعية الموجودة داخله لذلك يجب اتباع ما يلي عند حذف أحـــد الفـــهارس الفرعية .

1. الانتقال الى الفهرس الفرعي المطلوب حذفه بالأمر CD .
2. حذف محتويات هذا الفهرس الفرعي المطلوب باستخدام الأمر *.* del
3. حذف الفهارس الفرعية إن وجدت فيه .
4. الرجوع إلى الفهرس الذي يحتوي الفهرس المطلوب (أي الفـــهرس الأب) بواسطة الأمر ..CD
5. تنفيذ حذف الفهرس المطلوب بالأمر : اسم الفهرس المطلوب حذفه RD

مثال : أحذف الفهرس الفرعي Program مع العلم أن السواقة الحالية C .

نتيجة التنفيذ على الشاشة	أوامر الدوس المتتالية
A:\>	1) C:\> A: ↵
A:\> Program>	2) A:\> CD Program ↵
A:\> Program>program1>	3) A:\> Program>CD program1 ↵
حذف جميع الملفات الموجودة تحت الفهرس Program1	4) A:\>Program>Program1>del *.* ↵
حذف الفهرس الفرعي ss1 الموجود تحت الفهرس Program1	5) A:\>Program>Program1>rd ss1 ↵
A:\> Program>	6 A:\>Program>Program1>cd..↵
حذف الفهرس Program1	7) A:\>Program>RD Program1↵

• **الأمر DELTREE :**

وظيفته : حذف الفهرس الفرعي وفروعه وملفاته بخطوة واحدة وذلك باتباع ما يلي :

1- الانتقال إلى الفهرس الذي يحوي الفهرس المطلوب حذفه (الأب)

2- تكتب الأمر بالشكل التالي

اسم الفهرس المطلوب حذفه DELTREE <الفهرس الأب

مثال : المثال السابق ينفذ بالشكل التالي بعد الانتقال إلى الفهرس الفرعي Program

A:\> Program > DELTREE Program1

عندئذ يسأل الدوس إذا كنا فعلا نريد إزالته فنضغط Y وإلا نضغط مفتاح N.

هذا الأمر خطير إذا لا مجال للتراجع بينما في الأمر السابق تعدد الخطوات تجعل عملية الحذف بطيئة ويكون هناك مجال للتفكير في الحذف .

● الأمـــــر DISKCOPY :

وظيفة : نسخ محتويات القرص بأكمله إلى قرص آخر

صيغة :

C:\> DISKCOPY d1: d2: ↵

حيث d1 يمثل اسم السواقة التي تحوي القرص المراد النسخ منها (القرص الأصلي)

d2 يمثل اسم السواقة التي تحوي القرص المراد النسخ إليه (القرص المستقبل)

ملاحظات :

1- قبل البدء بعملية نسخ الأقراص يجب التأكد من أن القرص المستقبل خالياً مـــن أية بيانات وحجمه وسعته تطابق الحجم والسعة في القرص الأصلي .

2- ليس بالضرورة أن يكون القرص المستقبل مهيئاً بل إن الأمـــر نفســـه يهيئـــه أتوماتيكيا.

3-يجب حماية القرص الأصلي من إعادة الكتابة علية (بفتح البوابـــة الموجـــودة في الزاوية السفلي من القرص المرن) .

4-إذا كان لدينا سواقة واحدة للأقراص المرنة فإن العملية تتم بالشكل التالي:

نكتب الأمر كما يلي :

C:\> DISKCOPY A: A: ↵

سيبدأ النظام دوس بنسخ القرص الأصلي ويضع المعلومات المنسوخة في ذاكـــرة الحاسوب وحين تمتلئ الذاكرة ستظهر رسالة تطلب منـــك إدخـــال القـــرص المستقبل في السواقة .

- نسحب القرص الأصلي من السواقة ونضع القرص المستقبل مكانه ونضغـــط على مفتاح الإدخال ENTER عندئذ سيقوم الدوس بنقل البيانات الموجـودة في الذاكرة إلى القرص المستقبل .
- إذا كان حجم المعلومات الموجودة في القرص الأصلي اكبر من سعة الذاكـــرة عندئذ ستظهر رسالة تطلب منك إدخـــال القـــرص ثم نضغـــط ENTER

وسيقوم الدوس مرة ثانية بنقل بقية المعلومات إلى ذاكرة الحاسوب ثم يطلب إدخال القرص المستقبل لاستكمال عملية النسخ .

- وهكذا تستمر المبادلة إذا لزم الأمر لحين إتمام عملية نسخ القرص الأصلي وظهور رسالة يسأل فيها الدوس إذا كنا نريد نسخ قرص آخر أم لا فنضغطY في حالة الإيجاب ، ونضغط علىN في حالة النفي .

5- في حالة وجود سواقتين للأقراص المرنة في الحاسوب متشابهتين بالحجم والسعة فإن عملية النسخ تتم بسهولة وأسرع دون الحاجة إلي مبادلة الأقراص.

(5-5) نظام التشغيل Windows

(5-5-1) تعريف وميزات النظام ويندوز :

يعرف نظام التشغيل ويندوز على أنه نظام مواجهة رسومية Graphical User Interfaces يقوم بدور الوسيط بين المستخدم والحاسوب ، ويستخدم هذا النظام أشكالا ورموزا صغيرة تدعى الأيقونات Icons ، هذه الأيقونات تمثل تطبيقات معينة أو عمليات تتعلق بالتطبيقات بالإضافة إلى استخدامه لقوائم الأوامر Menus لتسهيل عمليات اختيار وتشغيل الأوامر عن طريق الفأرة Mouse

ولهذا النظام مزايا عديدة أهمها :

1- يتيح مشاهدة النص والرسوم على الشاشة تماماً كما ستظهر عند طباعتها.

2- يتيح تشغيل عدة برامج تطبيقية في أن واحد ، وكل برنامج في إطاره الخاص، على عكس نظام التشغيل دوس MS-DOS الذي يتيح تشغيل برنامج تطبيقي واحد في كل مرة .

3- جميع البرامج التطبيقية التي تعمل داخل النظام ويندوز تخضع لنفس القواعد والخطوات من حيث التشغيل والخروج من البرنامج والتحويل بين البرامج المختلفة وتكبير وتصغير ونقل الأطر ، مما يسهل تعلم البرامج التطبيقية الجديدة ، لذلك فالمهارات التي يكتسبها المستخدم بالعمل على النوافذ

الأساسية لنظام التشغيل ويندوز تسهل عليه الانتقال إلى البرامج التطبيقية المختلفة التي تعمل تحت هذا النظام .

4- يعطي إمكانية تناقل البيانات بين البرامج التطبيقية المختلفة التي تعمل تحت هذا النظام ، بكلمة أخرى إمكانية نقل وتحويل وربط النصوص والبيانات والرسوم بين البرامج التطبيقية المختلفة.

5- يعتمد نظام التشغيل ويندوز على استخدام الرسومات والأشكال في إدخال الأوامر للحاسوب بدلاً من كتابتها من خلال سطر الأوامر كما في نظام التشغيل دوس .

وقد أنتجت شركة Microsoft العديد من الإصدارات المختلفة لبرامج النظام ويندوز وكل إصدار مطور وأفضل من السابق في كثير من المتطلبات والمواصفات .

من تلك الإصدارات هي أنظمة التشغيل ويندوز Windows 3.x والتي كانت آخرها الإصدار 3.11 عام 1994 (Windows 3.11)

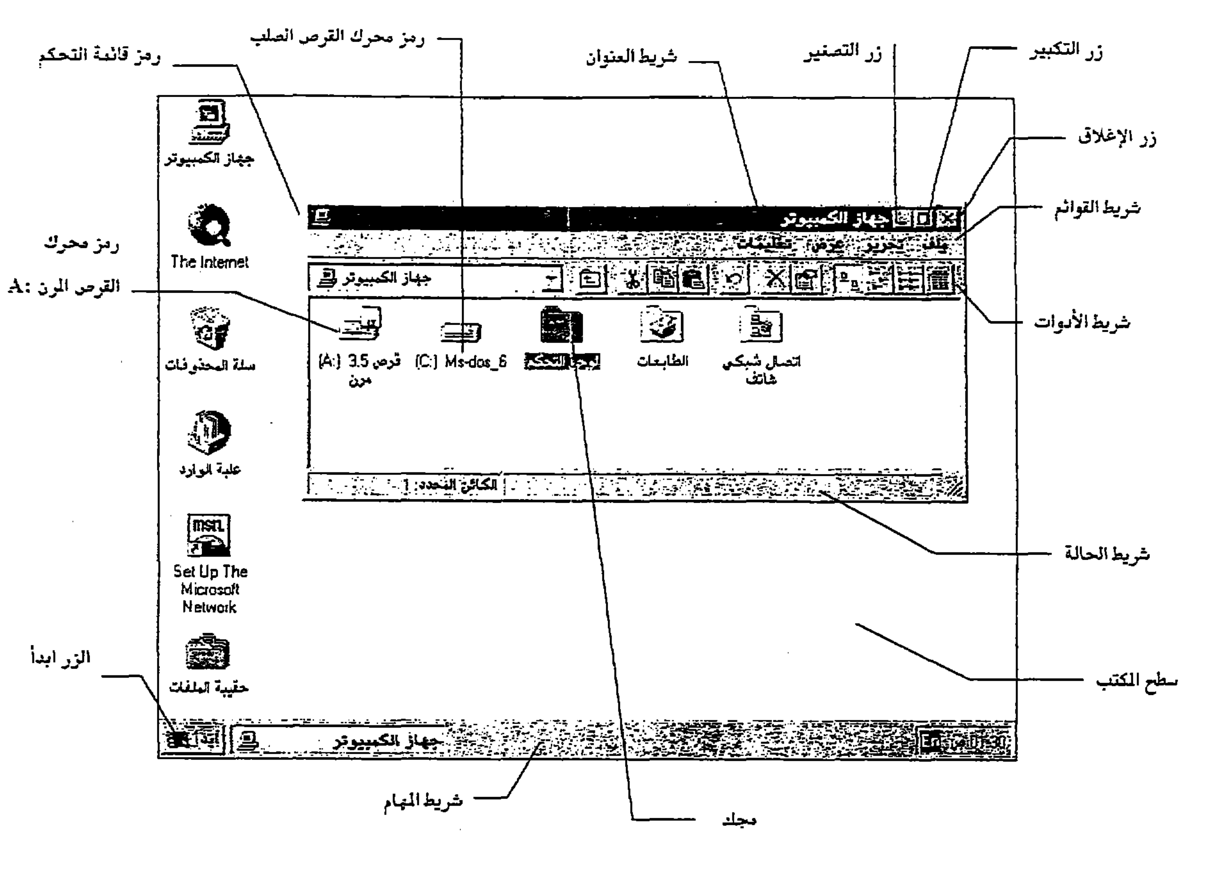
ثم تلى ذلك نظام ويندوز 95 الذي يعتبر نقله نوعية في عالم البرمجيات ، إذ يعتبر نظام تشغيل تعددي المهام له القدرة على دعم أنواع الشبكات ويسمح بالاتصال بالإنترنت ويعطي بنية جديدة لنظام الملفات تسمح بوصول أطوال الأسماء الى 256 رمزاً ، بالإضافة إلى أنه نظام تشغيل ذي 32 بتا يتمتع بواجهة استخدام مريحة وجذابة وفعالة في آن واحد ، وسوف نتطرق في الفقرات اللاحقة الى بعض المبادئ الأساسية للعمل على هذا النظام .

ثم ظهر ويندوز 97 (Windows 97) الذي عالج بعض المشاكل التي كان يعاني منها ويندوز 95 ، ثم ظهر ويندوز 98 (Windows 98) كتطوير لنظام ويندوز 95 ، في الحقيقة لا يوجد فرق يذكر بين ويندوز 98 ويندوز 95 ، فالشاشة الرئيسية والشاشات الفرعية في نوافذ 98 تشابه مثيلاتها في نوافذ 95 مع

بعض الإضافات الطفيفة ، لذلك فمستخدمي ويندوز 95 يمكنهم التعـــامل مـــع ويندوز 98 بكل سهولة وبدون أية مشاكل .

لقد جاء ويندوز 98 لتسهيل التعامل مع معدات الحاسـوب Hardware مثل :

1. تسهيل عمليات نقل البيانات من والى الذاكرة الرئيسية .
2. ضمان سرعة أكبر للجهاز .
3. توفير حيز أكبر للتخزين .
4. تسهيل عمليات التعامل مع شبكة الإنترنت والبريد الإلكتروني .
5. تصحيح أعطال الملفات بشكل أتوماتيكي حين حدوثها .
6. دعم مشغلات الأقراص الصلبة ذات السعة العالية .
7. التعرف على معدات الحاسوب المضافة بشكل افضل .
8. تقليل الأخطاء الناجمة عن المعدات والبرمجيات .
9. توفير قوائم مساعدة (Help) متقدمة وشاملة .



الشكل (5) نافذة التشغيل في ويندوز 95

(5-5-2) المبادئ الأساسية للتعامل مع النظام ويندوز 95

Controlling Windows 95

أولاً : تشغيل ويندوز 95 : Starting Windows 95

وذلك بالضبط على زر التشغيل Power ، عندئذ يظهر شعار Microsoft Windows الذي يبقى على الشاشة بضع ثوان ، ثم تظهر نافذة تدعى نافذة التشغيل كما في الشكل (5) ، تحوي هذه النافذة على أيقونات البرامج وشريط المهام وزر البدء ...

أما الأيقونات التي عادة تظهر عند تشغيل ويندوز 95 هي :

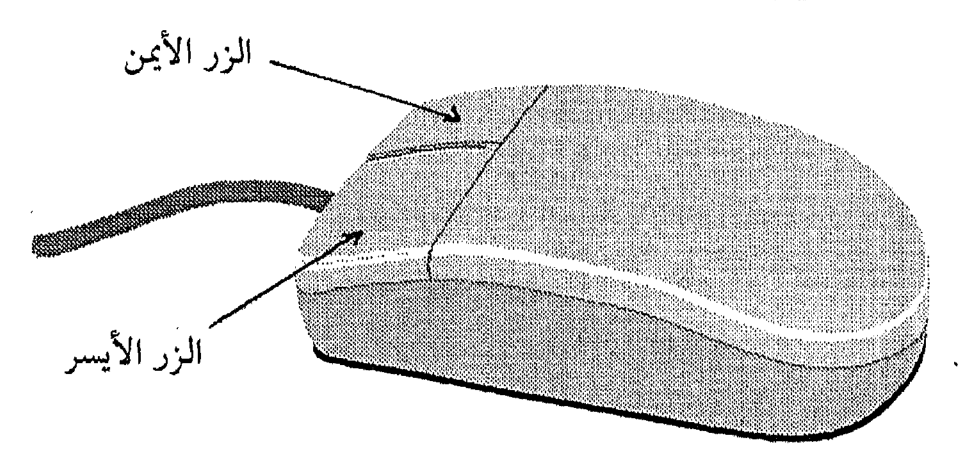
1. أيقونة جهاز الكمبيوتر My Computer : وتضم داخلها أيقونات فرعية أخرى :
   - ايقونة محرك القرص المرن A : تفيد في استعراض محتويات القرص المرن (A) .
   - أيقونة نظام التشغيل MS-DOS : للانتقال إلى شاشة الدوس .
   - أيقونة لوحة التحكم Control Panel: للتحكم بسرعة الفأرة وألوان الشاشات وشاشة التوقف ..
   - أيقونة الطابعات Printer : للتحكم بالطابعة المتصلة مع الجهاز .
   - أيقونة الاتصال الشبكي الهاتفي .
2. أيقونة الإنترنت The Internet : للاتصال بشبكة الإنترنت .
3. أيقونة Setup the Microsoft Network
4. أيقونة سلة المحذوفات Recycle Bin ، يتم داخلها حفظ الملفات التي يتم حذفها ، هذه الايقونة تعطي إمكانية استرجاع ملف محذوف سابقا ، وعادة تبقي الملفات المحذوفة داخلها ما لم نقوم بافراغ السلة .
5. أيقونة حقيبة الملفات My Briefcase .

6. بالإضافة الى ايقونات برامج تطبيقية أخرى ويتعلق ذلك بالمستخدم نفسه والبرامج التي تهمه بشكل دائم .

ثانياً : استعمال الفأرة Using Mouse

تقوم الفأرة بالكثير من المهام فهي الأداة الرئيسية فيه ، إذ أنها توفر على المستخدم عناء إدخال الأوامر من لوحة المفاتيح ، وتزيد من سرعة وفعالية العمل ونستطيع بالفأرة الوصول إلى كل زوايا الشاشة والتنقل بحرية في كل مكان مما يجعل منها أداة مثالية في نظم التشغيل ذات الواجهات الرسومية مثل النظام ويندوز ..

تتألف الماوس التي تستخدم مع ويندوز 95 من زرين (الزر الأيسر والزر الأيمن) أنظر الشكل (6) .



الشكل (6) أجزاء الفأرة

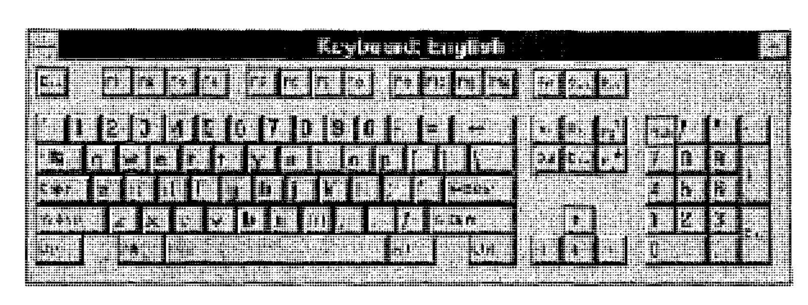
إن المصطلحات المستخدمة والعمليات التي تقوم بها الفأرة هي :

- الزر النشط : هو الزر الأيسر للفأرة .
- مؤشر الفأرة : وهو سهم صغير يتحرك على الشاشة عند تحريك الفأرة في أي اتجاه نريد ، ونستفيد منه في التأشير على مكان معين على الشاشة ، تدعي هذه العملية بالتأشير Pointing .

- النقر Clicking : ضغط الزر الأيسر للفأرة مرة واحدة و إفلاته بسرعة دون تحريك الفأرة على رمز أو أمر أو ايقونه معينة يقع عليها مؤشر الماوس وبالتالي تستخدم عملية النقر للتحفيز .
- النقر المزدوج Double Clicking يعني ضغط الزر الأيسر للفأرة مرتين متتاليتين على المكان الذي تقع عليه مؤشر الفأرة ، وتستخدم هذه العملية للتنفيذ .
- النقر والسحب Dragging يعني مواصلة الضغط على زر الفأرة الأيسر وتحريك الفأرة إلى مكان آخر من الشاشة ثم إفلات زر الفأرة ، ويستخدم من اجل نقل ايقونة أو نافذة معينة إلى مكان آخر على الشاشة .

ثالثاً : استعمال لوحة المفاتيح Using Keyboard

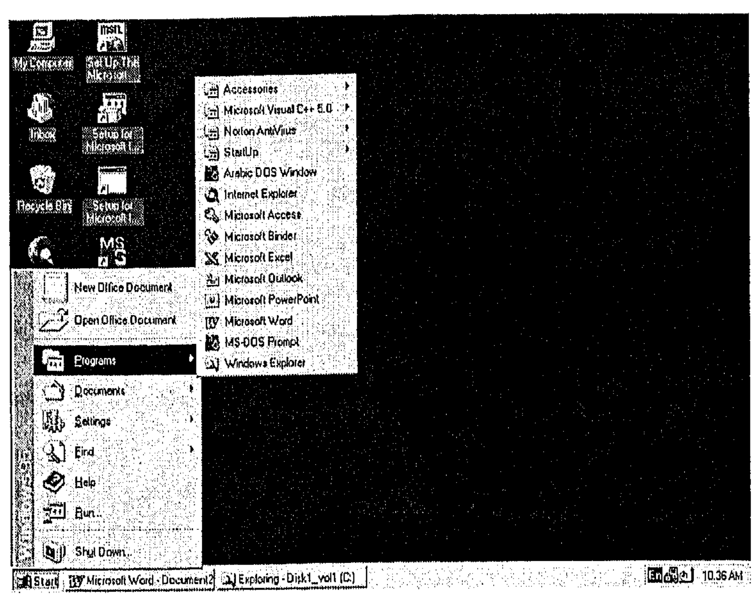
كما هو معروف فإن لوحة المفاتيح تعتبر من أهم وحدات إدخال البيانات إلى الذاكرة و تتضمن الحروف الأبجدية (العربية والإنجليزية) و حروف الترقيم والأرقام ،بالإضافة إلى مفاتيح خاصة تدعى مفاتيح وظيفية (Function Keys) ، أنظر الشكل (7) ، تقوم بمهمات حسب البرنامج التطبيقي ، فمثلاً مفتاح Shift مع أحد مفاتيح الأحرف يؤدي إلى طباعة الحرف العلوي من المفتاح.



الشكل (7) لوحة المفاتيح

رابعاً : استعمال زر " ابدأ " : **Using the Start Button**

يظهر زر البدء Start في اسفل نافذة التشغيل على شريط المهمة Task Bar أنظر الشكل (5) ، عند النقر عليه تظهر قائمة البدء التي تضم مجموعة من الأوامر التالية (الشكل (8)) :



الشكل (8) قائمة زر البدء start

أمر البرامج Programs : بالنقر عليه تظهر قائمة فرعية تبـــين التطبيقـــات المخزنة داخل الجهاز ويمكن تشغيل أي من التطبيقات بالنقر المزدوج عليها مـــن هذه الـــبرامج برامـــج وورد (Microsoft Word) ، برنـــامج اكســـل (Microsoft Excel) برنامج فوكس برو (Microsoft FoxPro)...

ملاحظة : أن وجود الاشارة أمام الأمر تعني أنه هناك قائمة فرعية تظهر عنـــد اختيار هذا الأمر .

ووجود ثلاثة نقاط امام الأمر ... يعني أنه يظهر عند اختيار هذا الأمـــر صنـــدوق حوار Dialog Box يعطي إمكانيات إضافية للتحكم وللسيطـــرة على تنفيـــذ الأمر المختار .

- أمر مستندات Documents : بالنقر عليه تظهر قائمة فرعية تبين المستندات (ملفات) تم العمل بها مسبقا ، ويتم فتح أي ملف بالنقر المزدوج عليه.
- أمر إعدادات Setting : بالنقر عليه تظهر قائمة فرعية تظم ثلاثة أوامر:

1. لوحة التحكم Control Panel : وظيفتها التحكم في اعداد الأيقونات، وتغيير اللغة والتحكم في سرعة النقر بالفأرة وشكل المؤشر وأزرار الفأرة بالإضافة الى تغيير الرقم السري Password والتحكم بألوان عناصر النوافذ،...
2. اعداد الطابعة Printers : وظيفته فحص حالة الطابعة وتوقيف الطابعة بشكل مؤقت واستئناف الطباعة ، ...
3. شريط المهام Task Bar : لتحديد من خلاله بعض الخصائص المتعلقة بشريط المهام من إخفاء أو تصغير أو تكبير أو تغيير مكانه أو أية خصائص اخرى ممكنة .

- أمر بحث Find : بالنقر عليه تظهر قائمة فرعية تظم أمرين :

- Files or Folder : مهمته البحث عن ملف معين أو البحث عن مجلد معين Folder .
- Clean Sweep

- أمر Help : بالنقر عليه يظهر ملف التعليمات المساعدة لارشاد المستخدم عن كيفية استخدام أمر معين أو تنفيذ مهمه معينة من مهام نظام ويندوز .
- أمر Run : لتشغيل برنامج أو ملف معين .
- الأمر Shut Down : بالنقر عليه يؤدي الى إنهاء العمل على ويندوز وهنا يجب أن تكون جميع النوافذ مغلقة قبل استخدام أمر Shut Down .

خامساً : التحكم بالنافذة Controlling a window

يظهر النظام ويندوز 95 كل معلوماته في داخل صناديق أطر على الشاشة تدعى بالنوافذ Windows ، وقبل أن نتمكن من التعامل مع معلومات أي نـــافذة لا بد من تعلم كيفية فتحها ، تصغيرها وتكبيرها ، نقلها إلى أي مكان على الشاشة وتغيير حجمها .

أ. فتح نافذة : Opening a window

- نضع مؤشر الفأرة على رمز " جهاز الكمبيوتر " My Computer
- ننقر زر الفأرة الأيمن ، تظهر القائمة السريعة .
- نختار الأمر " فتح " Open.
- تظهر نافذة شبيهه بالنافذة الرئيسية باسم My Computer تتألف من :

1. شريط العنوان : Title Bar : لإظهار اسم النافذة المفتوحة .
2. في الطرف الأيسر من شريط العنوان يوجد زر قائمة التحكـــم Control Menu يعطي للمستخدم إمكانيات التحكم بالنافذة المفتوحة ، عند النقـــر عليها بواسطة زر الفأرة الأيسر ، باستخدام أوامر مثل تصغير ، أو تكبير ، أو تحريك النوافذ . إذا نقرنا على زر قائمة التحكم بواسطة زر الفـــأرة الأيمـــن تظهر القائمة السريعة التي تظم أيضاً بعض الأوامر المفيدة .
3. الطرف الأيمن من شريط العنوان يوجد ثلاثة أزرار كل واحد يؤدي مهمـــه معينة عند النقر عليه بواسطة زر الفأرة الايسر :

Minimize Button	—	تصغير نافذة الى زر يظهر على شريط المهام .
Maximize Button	☐	يمكن تكبير النافذة بحيث تغطي كل الشاشة .

Restore Button		إعادة النافذة الى حجمها الاصلي،ويظهر هذا الزر عندما تكون النـــافذة في حالـــة الحجـــم الاقصى أي تغطي كامل الشاشة .
Close Botton	X	اغلاق النافذة .

4. تحت شريط العنوان يظهر شريط القوائم Menu Bar والذي يضم أسمـــاء عدة قوائم وكل قائمة، عند النقر عليها بزر الماوس الأيسر ، تضم مجموعـــة من الأوامر .

5. شريط الأدوات Tool Bar ويضم أزرار تمثل أوامر مهمات معينة بهـــدف توفير الوقت للمستخدم بدلاً من الأوامر الموجودة داخل القوائم .

6. شرائط التصفح Scroll Bar ، وتظهر أسفل الشاشـــة أو علـــى يســـارها مرسوم عليها أسهم وظيفتها إظهار القسم المخفي من محتويات النافذة عنـــد النقر على أحد الأسهم ،وهنا تشير أن شريط التصفح يظهر فقـــط عندمـــا يكون حجم محتويات النافذة اكبر من مساحة النافذة المفتوحة .

7. شريط المهام Task Bar ويظهر في أسفل النافذة ويمكن أن يضم ميقاتيـــة الوقت على الجهة اليمنى وزر ابدأ (Start) على الجهة اليسرى بالإضافـــة الى اسهم النافذة المفتوحة أو الأوامر المشار إليها بواسطة الماوس أو المهمة التي يقوم بها الحاسوب .

8. الرمز (الأيقونات) Icon وهي رسوم صغيرة لتطبيقات مصغـــرة إلى الحـــد الأدنى ويتم فتحها بالنقر المزدوج عليها .

ب. نقل النافذة وتغيير حجمها Moving and Resizing

1. نقل النافذة : نضع مؤشر الفأرة على شريط العنوان Title Bar ثم نضغـــط على زر الفأرة الأيسر مع مواصلة الضغط والسحب إلى موقع جديد ، عندئـــذ

سيتحرك إطار النافذة مع حركة مؤشر الفأرة ، وعند تحرير زر الفأرة تستقر النافذة مع محتوياتها في مكانها الجديد .

2. تغيير حجم النافذة : نضع مؤشر الفأرة على أحد أضلاع (إطار) النافذة عندئذ يتغير شكل مؤشر الفأرة ويصبح على شكل سهمين متعارضين ↔ ، نضغط على زر الفأرة الأيسر مع السحب إلى اليمين أو اليسار حسب الحجم الذي تريد .

ويمكن أيضا بنفس الطريقة سحب إحدى زوايا النافذة لتحريك ضلعين في نفس الوقت سواء للتكبير أو التصغير .

سادساً : ترتيب النوافذ والرموز على سطح المكتب

Arranging Windows and Icons on the desktop

إن ترتيب النوافذ على سطح المكتب يفيد في تنظيم العمل ، ويعطي رؤية واضحة للعناصر لا سيما أن كثرة النوافذ على سطح المكتب تجعل من الصعب أيجاد ما نبحث عنه بسهولة .لذلك ولتقدم فعالية عمل أكبر في ويندوز 95 نتبع مايلي :

1. نضع مؤشر الفأرة على شريط المهام Task Bar
2. ننقر زر الفأرة الأيمن فتظهر قائمة تظم الأوامر التالية :

- تتالي Cascade
- تجانب أفقي Tile Horizontally
- تصغير كافة الإطارات Minimize All Windows
- تجانب عامودي Tile Vertically
- تراجع تتالي Undo Tile
- خصائص Properties

نختار أحد الأوامر السابقة حسب ما نريد ترتيب النوافذ المفتوحة على الشاشة.

أما لترتيب الرموز المبعثرة على سطح المكتب فيتم بالشكل التالي :

1. ننقر زر الفأرة الأيمن على أي منطقة خالية من سطح المكتب .
2. تظهر قائمة أحد أوامرها ترتيب الرموز Auto Arrange
3. نختار الأمر ترتيب الرموز فيتم ترتيب الايقونات بشكل تلقائي

سابعاً : استعمال القوائم Using Menus

معظم النوافذ في ويندوز 95 يحتوي على شريط القوائم Menus Bar الذي يضم مجموعة من أسماء القوائم مثل :

ملف File ، تحرير Edit ، عرض View ، خيارات Option ، تعليمات Help ، وغير ذلك حسب التطبيق المفتوح .

وكل قائمة من القوائم تضم مجموعة من الأوامر كل منها له مهمة معينة ، وهنا يجب أن نشير إلى بعض المصطلحات التي يمكن أن نشاهدها عند التعامل مع القوائم والتي ندرجها في الجدول التالي :

ما نشاهده في القائمة	المعنى
1. ثلاثة نقاط بعد الأمر (...)	عند اختيار الأمر سيظهر صندوق حوار يحوي المزيد من الخيارات .
2. أمر لون كاشف (إضاءة خافته)	لا يمكن تنفيذه في الوقت الحالي لأسباب تتعلق بطبيعة لأمر نفسه
3. مثلث بعد الأمر على شكل سهم	عند اختيار الأمر سيظهر قائمة ثانوية أخرى تحوي المزيد من الأوامر
4.علامة التحقق قبل الأمر (check mark)	عند ظهورها قبل الأمر تعني أن الأمر فعّال ويمكن لأكثر من أمر في المجموعة الواحدة أن يحمل علامة التحقق ويتم إظهارها بالنقر بزر الماوس الأيسر على الأمر، أما إخفاءها فيتم بالنقر مرة أخرى على الأمر
5.نقطة بلون غامق قبل الأمر (•)	تعني أن الأمر فعال ، ولا يمكن لأكثر من أمر في المجموعة الواحدة أن يحمل نقطة ولوضع النقطة ننقر زر الماوس الأيسر على الأمر ، أما إخفاءها

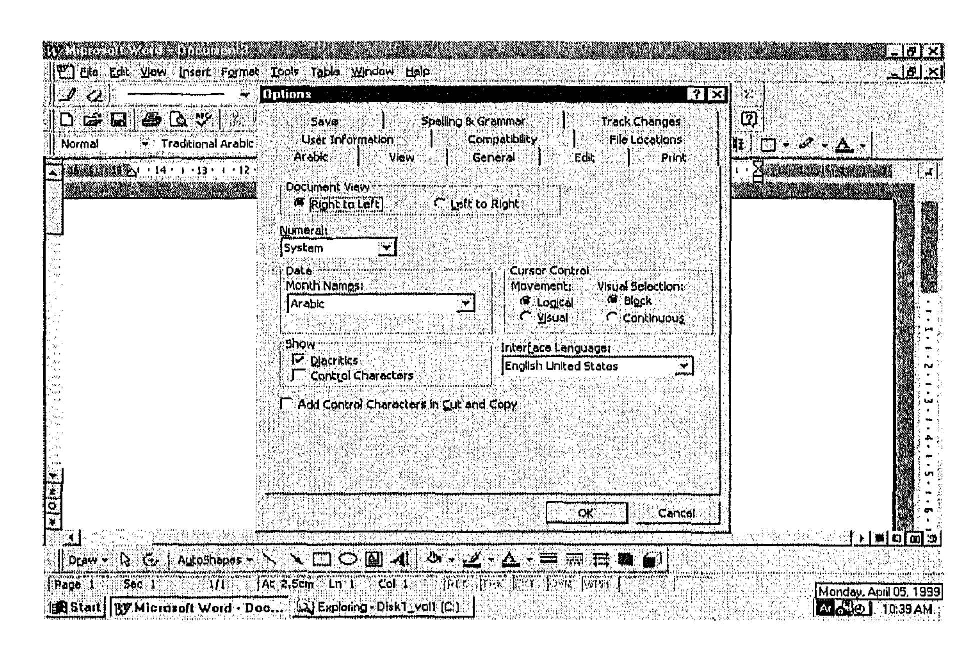
	فيتم بالنقر على أمر أخر ضمن المجموعة.
6. اسم مفتاح ائتلاف بعد الأمر	يمكن تنفيذ الأمر بواسطة المفتاح المشار اليه مثل ctrl+p للطباعة بدلا من فتح القائمة في كل مرة.

ثامناً : استعمال صناديق الحوار Using a Dialog Box

صندوق الحوار Dialog Box يظهر بشكل مستطيل ، يحتوي على عدد من العناصر والخصائص الإضافية التي يمكن اختيار بعضها أو كلها حسب ما تتطلب الحاجة ، وأغلب الصناديق في النظام التشغيل ويندوز تحمل عناصر مشتركة يتم التعامل معها بطرق متشابهة ، أنظر الشكل (9) والتي ندرجها في الجدول التالي:

اسم العنصر	شكل العنصر	الشرح
1.مفتاح الخيارات Options buttons	⊙	مفاتيح دائرية بيضاء يؤدي اختيارها بالنقر عليها بزر الفأرة الأيسر الى ظهور نقطة سوداء داخل الدائرة البيضاء،وهذا يعني أن الخيار فعال وهنا يجب أن نلاحظ أنه يمكن تفعيل خيار واحد فقط من ضمن كل مجموعة من مجموعات صندوق الحوار .
2.صناديق التحقق Check Boxes	☐	مربعات صغيرة أمام الخيارات، يتم تفعيل الخيار بالنقر بزر الفأرة الأيسر مرة واحدة عليه وتظهر إشارة التحقق داخل المربع ولازالة التفعيل ننقر مرة أخرى على الخيار.ملاحظة: يمكن تفعيل أكثر من مربع ضمن المجموعة الواحدة على عكس مفاتيح الخيارات .

الأداة	الشكل	الوصف
3.صناديق اللوائح المنسدلة List Boxes		وهي لائحة تحتوي على مجموعة من العناصر (مثل الملفات) والتي يمكن اختيار إحداها ولفتح اللائحة ننقر بزر الفأرة الأيسرعلى السهم الموجود على يمين اللائحة،ويمكن تصفح محتويات اللائحة بتدويرها الى الاعلى أو الاسفل باستخدام الأسهم التي تظهر على أحد جانبي اللائحة
4.الصناديق النصية Text Boxes		صناديق يمكن إدخال مقاييس فيها ، مثل إدخال حجم الخط ، عدد النسخ، اسم الملف ...
5.أزرار الأوامر Command Buttons	OK CANCEL HELP OPTIONS..	وهو شكل مربع أو مستطيل مكتوب عليه: ok أي الموافقة على الخصائص والخيارات المختارة cancel إلغاء الأمر والإبقاء على الخيارات القديمة ويمكن أن يكون مكتوب عليها أوامر أخرى تؤدي عند اختيارها إلى صندوق حوار آخر أو إنجاز مهمة معينة مثل زر Options
6. الألواح Tabs		عبارة عن صفحات ضمن صندوق الحوار تحتوي على معلومات وخيارات محددة وعادة تكون هذه الألواح في أعلى الصندوق واللوح الفعال يكون بشكل مستطيل نافر ، لاختيار صندوق آخر ننقر بزر الفأرة الأيسر على أسم اللوح الآخر فنجد أن الخيارات قد تغيرت بخيارات أخرى تناسب اللوح المفتوح.



الشكل (9) أحد نماذج صناديق الحوار

(5-5-3) التعامل مع المجلدات والملفات

Working With Folders and Files

(5-5-3-1) مقدمـــة :

لقد تطرقنا في الفقرة (5-4) إلى نظام التشغيل MS-DOS وتعرفنا على وظائفه ومهامه وكيفية التعامل مع الملفات والفهارس وإجراء العمليات الضروريـــة عليها ، أما في هذه الفقرة فسنتعرف على كيفية التعامل مع الملفـــات والمجلـــدات (الفهارس) ضمن نظام تشـــغيل مطـــور وهـــو نظـــام التشـــغيل وينـــدوز 95 (WINDOWS 95) ، الذي يسهل الكثير من العمليـــات وقبـــل أن ندخـــل بالتفاصيل سنقدم لمحة بسيطة عن MS-DOS نلخص من خلالها أهم المـــهمات والوظائف لهذا النظام وعلاقة هذه الوظائف بنظام التشغيل ويندوز .

يتألف كل قرص مرن أم صلب أم مضغوط CD من فـــهرس رئيســـي توضع عليه الملفات والفهارس الفرعية ، وتتألف الفهارس الفرعية بدورهـــا مـــن

فهارس فرعية أخرى وملفات ، والفرق بين الملف والفهرس هو أن الفهرس شـكل تنظيمي يتم تجميع فيه الملفات والفهارس الفرعية المنتمية الى موضـوع محـدد أي مكان تخزيني شبيه بقرص ، أما الملف فهو عبارة عن مجموعة مـن البيانـات ذات علاقة معينة تخزن تحت اسم معين .

يمكن التميز بين العمليات التي تنجز على الفهارس وبين التي تجري علـى الملفات حيث نجد أن أهم عملية يتعرض لها المستخدم عند العمل علـى ملفـات لبرامج هي تشغيل البرنامج بكتابة أسم الملف عند محث نظام التشغيل MS-DOS ثم الضغط على ENTER واهم عملية تنجز على الملفات النصية هي فتـح هـذه الملفات بواسطة محررات النصوص لتعديلها أو الإضافة عليها أو قراءتها... الخ ، أمـا بالنسبة للفهارس فأن أهم عملية هي الدخول إلى الفهارس للوصول الى محتوياتـه باستخدام الأمر (CD) يليه أسم الفهرس الفرعي ، ثم نضغط ENTER بالإضافة الى الأوامر الأخرى التي تساعد على نقل الملفات من فهرس الى فـهرس آخـر أو نسخها أو حذفها أو ... الخ ، وكل هذه الأوامر يجب حفظها ليتم إدخالها عنـد محث النظام DOS ثم الضغط على مفتاح الإدخال ENTER.

مع ظهور أنظمة التشغيل ويندوز ، أصبح التعامل مع تلك الأوامر اسـهل وأسرع حتى أن الكثير من مستخدمي الحاسوب يفضلون التعامل مع نظام التشـغيل ويندوز واعتبروا أن الدوس طراز قديم وعقيم الفهم والتعامل .

لقد اصبح الفهرس الفرعي Subdirectory في نظام ويندوز 95 يدعى مجلداً Folder ، ويتم الانتقال إليه بالنقر عليه نقرة واحدة بواسطة الفأرة، أمـا فتحه واستعراض محتوياته فيكون بالنقر المزدوج عليه ، كما سنشرح ذلك لاحقـاً، أما بالنسبة للملفات فيؤدي النقر المزدوج بواسطة الفأرة على ملـف تنفيـذي لتطبيق معين الى تشغيل ذلك التطبيق ، بينما يؤدي النقر المزدوج على ملف نصـي الى جلب التطبيق الذي كتب فيه ذلك المستند وفتحه . بالإضافة الى ما سبق فـإن

ويندوز 95 يقدم خدمات اخرى تساعد على تنظيم العمـــل علـــى المجلــــدات والملفات مثل خاصية البحث عن ملف أو مجلد وتحديد موقعه والانتقال إليه .

(5-5-3-2) طرق التعامل مع الملفات والفهارس :

يقدم النظام ويندوز 95 طريقتين للتعامل مع الفهارس والملفات :

الطريقـــة الأولى : وتتمثـــل باســـتخدام ايقـــونـــة " جـــــهاز الكمبيوتــــــر" "My Computer" الموجودة على سطح المكتب كما يلي :

1. النقر المزدوج على ايقونة جهاز الكمبيوتر "My Computer" تظهر نـــافذة تحوي ايقونات مختلفة بالإضافة إلى ايقونات محركـــات الأقـــراص (ســـواقات الأقراص) المتوفرة على الحاسوب مثل A,B , ومحرك الأقراص المضغوطة .
2. نقوم بالنقر المزدوج على أحد رموز محركات الأقراص ، يؤدي ذلك الى ظـــهور نافذة تستعرض محتويات ذلك المحرك ، وتسمح بإجراء العمليات اللازمة فالنسخ والنقل والحذف وغير ذلك .

الطريقـــة الثانيـــــة : فتتمثـــــل باســـــتخدام ، مستكشـــــف وينـــــدوز "Windows Explorer" كما سنشرح في الفقرة التالية :

الاختلاف بين الطريقتين الأولى والثانية هو أن استخدام مستكشف يسمح بمشاهدة جميع الملفات والمجلدات على الفهرس الرئيسي ضمن القســـم اليســـاري لنافذة المستكشف ، بينما يعرض في القسم الأيمن من النافذة محتويات المجلد المختـــار ، بينما يؤدي استخدام طريقة " جهاز الكمبيوتر " إلى فتح نافذة جديدة لعــــرض محتويات المجلد مما يتسبب بامتلاء الشاشة بالنوافذ ، بالإضافـــة إلى أن مستكشـــف النوافذ هو عبارة عن مدير الملفات في ويندوز 3.x ، بل يعتبر نسخة مطورة عنه.

(5-5-3-3) : استخدام مكتشف النوافذ
Using The Windows Explorer

مستكشف النوافذ هو أداة لتنظيم الملفات والفهارس داخل الحاسوب ، حيث يمكننا من عرض محتويات الأقراص والفهارس ، ونقل ونسخ وحذف الملفات وتهيئة الأقراص ، وإنشاء فهارس جديدة وإعادة تسمية الفهارس وتشغيل التطبيقات ...الخ.

إن العمل على مستكشف النوافذ في ويندوز 95 أو ويندوز 98 يؤدي لنفس الغرض والمهام التي نقوم بها باستخدام نظام التشغيل MS-DOS ، بالإضافة إلى أسلوب التعامل بين المستخدم والحاسوب أفضل وأسهل لذلك سنقدم في الفقرة القادمة بعض أوامر الدوس بأسلوب ويندوز 95 ، بما يتناسب مع مناهج هذه المادة ، وبما يلبي حاجة القارئ في البدء للعمل على نظام التشغيل ويندوز 95 أو 98 .

1.تشغيل مستكشف النوافذ :

- النقر على زر ابدأ (START) ، تظهر قائمة ابدأ .
- النقر على أمر البرامج Programs ، تظهر قائمة فرعية بالملفات والبرامج .
- النقر على Windows Explorer مستكشف النوافذ ، عندئذ تظهر النافذة المطلوبة كما في الشكل (10) .

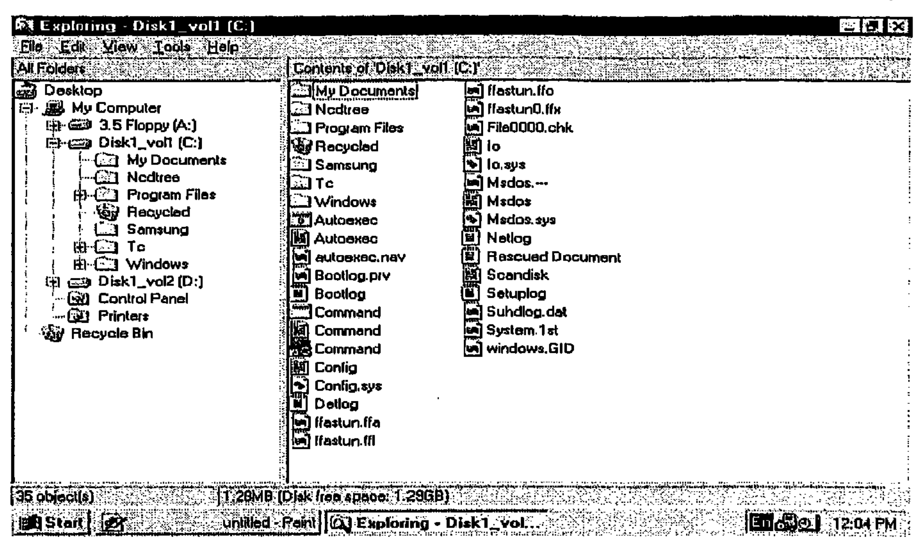
تتألف نافذة مستكشف النوافذ كما في أي نافذة من :

1.شريط العنوان Title Bar والذي يحوي في الطرف الأيمن ثلاثة أزرار :
(تصغير [_] ، تكبير [□] ، إغلاق [X])

2.شريط القوائم Menus Bar ويظم القوائم :

File الملف
Edit تحرير

View عرض
Tools أدوات
Help التعليمات .



الشكل (10) نافذة مستكشف النوافذ

3. شريط الأدوات : الذي يضم أزرار تستخدم اختصارا بدل أوامر موجـــودة في القوائم مثل القص، النسخ ، اللصق ، بالإضافة الى صندوق يظهر فيه الفـــهرس (المجلد) المختار .

4. سطح المكتب أو المنطقة المعروض فيها الفهارس وتقسم إلى قسمين الأيسر يضـــم جميع المجلدات (الفهارس) All Folders. بما أنهـــا محركـــات الأقـــراص ، والقسم الأيمن يضم الملفات والفهارس الفرعية الموجودة داخل المجلد (الفـــهرس) المختار .

2 – فتح نافذة فهرس جديد بواسطة رمز محرك الأقراص :

بالنقر المزدوج على رمز محرك الأقراص (A) تظهر جميع الملفات والفـهارس الموجودة في محرك الأقراص المختار تحت العنوان (Name) في القسم الأيمن مـــن النافذة وعندها يمكن فتح أي ملف بالنقر المزدوج على الملف المطلوب .

3- تهيئة الأقراص المرنة Formatting

ويعني تجهيزها وفحصها من الأعطال قبل التخزين عليها أما إذا كان القـرص يحوي معلومات مخزنة فإن عملية التهيئة تؤدي الى حذف جميع المعلومات على ذلك القرص ، ولتهيئة القرص المرن نتبع ما يلي :

1. نضع القرص المرن في محرك الأقراص المرنة (A).
2. ننقر بزر الفأرة الأيسر على رمز محرك الأقراص (A).
3. ننقر بزر الفأرة الأيمن ، فتظهر قائمة تحوي مجموعة من الأوامر .
4. ننقر الأمر Format عندئذ يظهر صندوق حوار باسم Format .
5. من صندوق القائمة المنسدلة (Capacity) نختار حجم القرص المـــراد تهيئته نختار الحجم 1.44 MB .
6. ننقر على الزر Full ليتم شطب محتويات القرص كليا .
7. ننقر على زر Start ليبدأ الحاسوب بتهيئة القرص .

4- ترتيب الملفات Arrange Files

1. ننقر زر الفأرة الأيمن على القسم الأيمن من النافذة فتظهر قائمة جديــدة تضم مجموعة من الأوامر .
2. نختار الأمر Arrange Icons ، عندئذ تظهر قائمة فرعية جديدة تحوي عدة أوامر .
3. لفرز المجلدات والملفات حسب الترتيب الأبجدي لملحق الأسم (اللاحقـة) نختار الأمر "By Type" الفرز حسب النوع .
4. لفرز المجلدات والملفات حسب حجمها تنازليا نختار الأمر "By Size" الفرز حسب الحجم .
5. لفرز المجلدات والملفات حسب تاريخ إنشائها بدأً من التـــاريخ الأخــير "By Date" الفرز حسب التاريخ .

5 – تحديد (اختيار) مجلد أو ملف Selecting Files Or Folders

ويتم ذلك بالنقر بزر الفأرة الأيسر على أسم الملف أو المجلد المراد اختياره وفي حالة تحديد أكثر من ملف عندئذ نضغط على المفتاح Ctrl وزر الفأرة الأيسر على الملفات المراد تحديدها .

6 – إنشاء فهرس (مجلد) Create Folder

لإنشاء مجلد جديد داخل مجلد آخر موجود نتبع ما يلي :

1. تحديد المجلد المراد إنشاء مجلد فرعي داخله .
2. نفتح القائمة File من شريط القوائم .
3. نختار الأمر New ، تظهر قائمة فرعية .
4. نختار الأمر Folder فيظهر صندوق حوار .
5. ندخل الأسم الذي نريده للفهرس ضمن الحقل المخصص ثم نضغط على زر ok

7– إعادة التسمية Renaming :

من أجل تغير اسم أحد المجلدات أو الملفات نتبع ما يلي :

1. تحديد الملف أو المجلد المراد تغير اسمه .
2. نفتح القائمة File من شريط القوائم .
3. نختار الأمر Rename عندئذ يظلل الأسم المراد إعادة تسميته .
4. نطبع الأسم الجديد ثم نضغط مفتاح الإدخال Enter من لوحة المفاتيح.

8– حذف ملف أو مجلد Delete File Or Folder :

لحذف ملف أو مجلد نتبع ما يلي :

1. نحدد الملف أو المجلد المراد حذفه .
2. نفتح القائمة File من شريط القوائم .

3. نختار الأمر Delete عندئذ تظهر رسالة لتأكد الحذف .

4. نختار الزر (ok) في حالة حذف ملف واحد أو مجلد واحد والزر (Yes To All) في حالة حذف عدة ملفات أو مجلدات .

9- نسخ الملفات أو المجلدات Copying Files Or Folders :

1. نحدد الملف أو المجلد المراد نسخه (عمل نسخة منه) .
2. نفتح القائمة Eidet من شريط القوائم .
3. نختار الأمر Copy .
4. ننقر عند المكان الذي نريد وضع النسخة من الملف أو المجلد فيه .
5. نضغط مفتاح الإدخال Enter من لوحة المفاتيح .

10- نقل الملفات أو المجلدات Moving Files Or Folders :

1. نحدد الملف أو المجلد المراد نقله .
2. نفتح القائمة Edit من شريط القوائم .
3. نختار الأمر Cut .
4. ننقر عند المكان الذي نريد وضع الملف أو المجلد المنقول .
5. نضغط مفتاح الإدخال Enter من لوحة المفاتيح .

11- إغلاق مكتشف النوافذ Ending Explorer :

وهنا لدينا أسلوبين :

1. نختار الأمر Close من القائمة File .
2. النقر على زر الإغلاق الموجود في الجهة اليمنى من شريط العنوان .

(6-5) البرمجيات التطبيقية الجاهزة : PACKAGES

هناك العديد من البرامج التطبيقية الجاهزة كل منها يلبي احتياجات معينة للمستخدم حسب مجال عمله واختصاصه ، ولا تحتاج هذه البرامج من المستخدم معرفة متعمقة بعلم الحاسوب أو حتى تركيب الحاسوب ومكوناته المادية ، بل

يكفي تعلم المستخدم كيفية التعامل مع البرنامج بحد ذاته وكيفية الدخـول إليـه واستخدام ميزاته المختلفة .

سنقدم في هذه الفقرة تعريف ببعض هذه البرامج الشائعة الاستخدام لـدى مختلف المستويات الاجتماعية ، ونترك التفاصيل الجزئية للعمل على هذه الـبرامج واستثمارها لمادة برمجيات الحاسوب الشخصي (1) وبرمجيات الحاسوب الشـخصي (2) والإحصاء المحوسب .

(5-6-1) برامج معالجة النصوص Word Processors

تعتبر برامج معالجة النصوص من برامج الحاسوب التطبيقية الأكثر انتشـاراً والهامة في تكنولوجيا المعلومات ، والمقصود من معالجة النصـوص هـو تسـخير مكونات الحاسوب وبرمجياته في إدخال نص كتابي الى نظام الحاسوب ، وتخزينـه ، التعديل عليه، عرضه على الشاشة بأشكال مختلفة ، استرجاعه وقـت الحاجـة ، تنسيقه بالشكل المناسب ، طباعته ، وهناك العديد من برامج معالجة النصوص مثـل : Microsoft Word , Write , Works , Word Perfect ... وغيرهـا .
وبرنامج مايكروسوفت وورد Microsoft Word اكثر هذه البرامج شيوعاً ولهذا سوف نتناول بعض ميزاته :

أولاً : يتم تشغيله من داخل نظام التشغيل ويندوز ، القائمـة Start الموجـودة اسفل الشاشة ثم النقر على الأمر برامـج Program ، ثم النقـر علـى Microsoft Word عندئذ تظهر نافذة الوورد انظر الشكل (11) .

ثانياً : العمليات الرئيسة لبرنامج مايكروسوفت وورد :

1. إنشاء نص Greating :

يتم إنشاء نص باستخدام لوحـة المفـاتيح Keyboard ، بعـد النقـر على الأمر New من القائمة File الذي يعطي إمكانيـة إنشـاء مسـتند

(صفحة للكتابة) يظهر فيها مؤشر صغـــير Cursor في بدايتـــها ، نقـــوم بإدخال الكلمات من لوحة المفـــاتيح عنـــد المؤشـــر ويتـــم الانتقـــال إلى السطر التالي تلقائياً عند وصول المؤشـــر إلى نهايـــة الســـطر دون الحاجـــة إلى الضغط على مفتاح الإدخال Enter، وبعـــد الانتـــهاء مـــن الكتابـــة يمكن استخدام الأمر تخزين Save مـــن القائمـــة File لتخزيـــن الملـــف على أحد وسائط التخزين المساعدة باســـم معـــين.

2. التحكم بـــالهوامش Indentation :

برنامج الوورد يعطي إمكانيـــة التحكـــم بـــالهوامش اليمـــنى و اليســـرى ورأس الصفحة واسفل الصفحة حسب مـــا تقتضـــي الحاجـــة وحســـب نوع الورق الذي سيستخدم للطباعة ، بالإضافـــة الى إمكانيـــة التحكـــم بالتباعد بـــين الفقـــرات Spacing والتبـــاعد بـــين الأســـطر line Spacing.

3. التعديل على النص المخـــزن Editing :

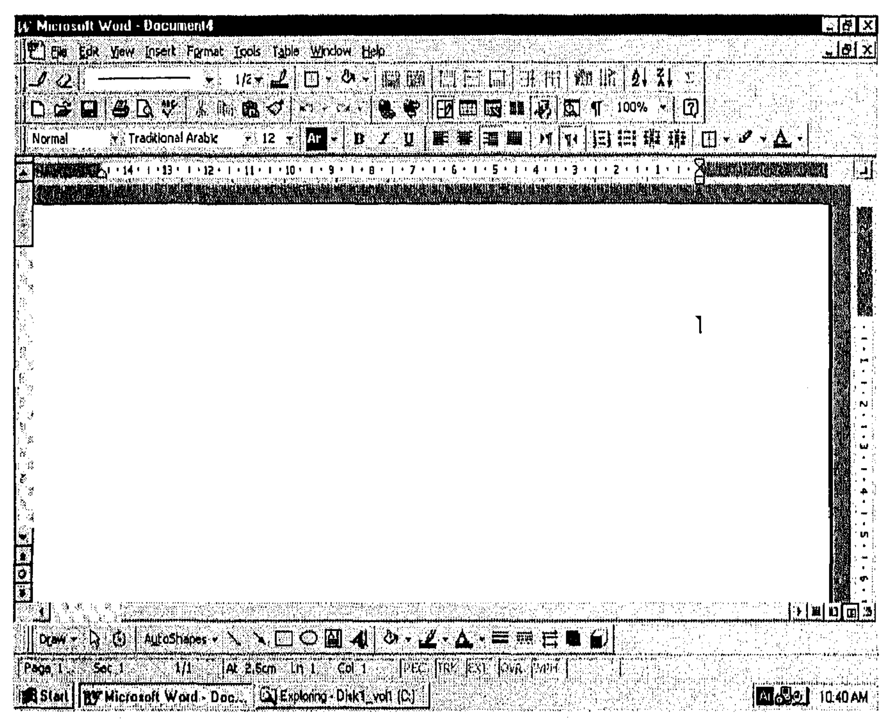
تعتبر إمكانية التعديل على النص من أقوى ميزات برنـــامج الـــوورد الـــتي تجعله يتفوق على وسائط الكتابة التقليدية مثـــل الآلات الكاتبـــة ، رغـــم أن التعديل متاحـــاً أثنـــاء الكتابـــة ، إلا أن المســـتخدم يســـتطيع أيضـــاً الرجوع الى النص الذي انتهى من إدخاله في أي وقـــت وعرضـــه علـــى الشاشة وإجراء التعديلات اللازمة ، وهـــذا يتيـــح للمســـتخدم مراجعـــة النص عدة مرات والتأكد مـــن عـــدم وجـــود أخطـــأ قبـــل الطباعـــة . بالإضافة إلى ميزات أخرى يمكن اســـتخدامها وهـــي نقـــل أو نســـخ أي مقطع من النص الى مكان آخر في نفس المستند أو إلى مســـتند آخـــر.

4. التحكم بحجم ونوع الخط **Controlling Font Kind & Size**

يتوفر في الوورد أنواع مختلفة من الخطوط العربية والأجنبية ، الأمر الـــــذي يتيح للمستخدم تغيير النص المكتوب أو المدخل بأي نوع مـــــن الخطــوط بالإضافة الى إمكانية تغيير حجم الخط في أي مكان من النص .

5. التدقيق الإملائي للنص : **Spelling** :

حيث يقوم الحاسوب باستخدام خاصية التدقيق الإملائي بمقارنة كل كلمـــة من النص بالكلمات المخزنــــة في قـــاموس الـــوورد Dictionary ، وإذا وجدت كلمة مختلفة عن نظيرها في القاموس يقدم الوورد عـــدة خيـــارات لتصحيحها ، إذ أن قاموس الوورد يحتوي على عشـــــرات الألـــوف مـــن الكلمات الشائعة الاستخدام ، مع إمكانية إضافة كلمات جديدة إليه .



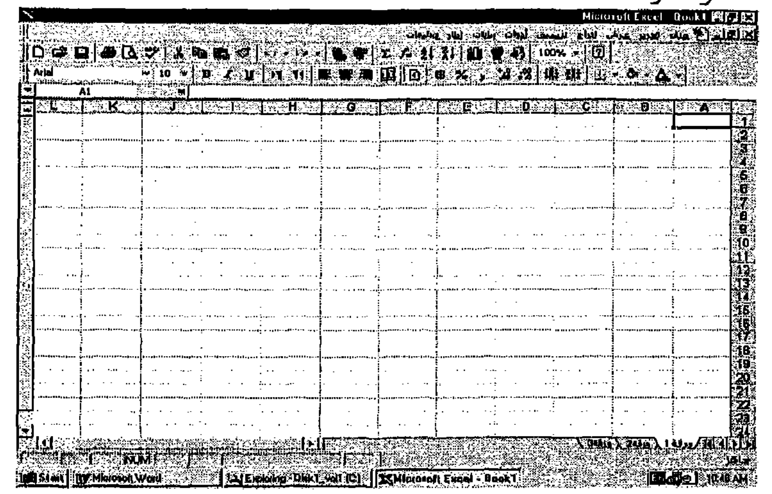
الشكل (11) نافذة مايكروسوفت وورد **Microsoft Word**

(2-6-5) الجداول الإلكترونية Spread Sheets

الجدول الإلكتروني هو جدول مؤلف من عدد كبير من الأعمدة والصفوف، تعرض الشاشة جزءاً صغيراً منه ، إلا أن المستخدم يمكنه التعامل مع الجزء المخفي عن طريق أزرار التصفح ، من وظائف الجداول الإلكترونية : تخزين البيانات والقيام بالعمليات الحسابية والتحليلات الإحصائية وإنشاء الرسوم البيانية لهذه البيانات باستخدام أوامر سهلة الاستعمال .

أول برامج الجداول الإلكترونية التي تم تصحيحها هو برنامج (Visual C) 1979 ثم ظهرت فيما بعد برامج كثيرة ومتطورة إلى أن أصبحت في الوقت الحالي تجمع بين خصائص الجداول الإلكترونية وخصائص برامج معالجة النصوص وبرامج إدارة قواعد البيانات من هذه البرامج ، برنامج (Ability Plus) وبرنامج (Louts 123) وبرنامج (Qwattropro 3) وبرنامج (Supercalc 5) وبرنامج (Excel) .

ويعتبر برنامج اكسل Excel أكثر البرامج استخداما في المؤسسات والشركات التجارية بسبب سهولة العمل عليه وميزاته التي تلبي حاجة المستخدم في المؤسسات والشركات .



الشكل (12) نافذة ميكروسوفت اكسل Microsoft Excel

أهم ميزات برنامج اكسل :

1. يتم تشغيله من داخل نظام التشغيل ويندوز ، القائمة Start الموجودة في اسفل الشاشة ثم النقر على الأمــــر برامـــج Program ، ثم النقـــر علـــى Microsoft Excel عندئذ تظهر نافذة ميكروسوفت اكسل انظر الشـــكل (12)
2. الاحتفاظ بالبيانات التي ندخلها الى ملفاته وإمكانية استرجاعها وقت الحاجة .
3. سهولة التحكم بالبيانات من ناحية مظهرها وإنجاز العمليات الحسابية اللازمـــة عليها مثل الضرب – الجمع – الطرح – القسمة ، و العمليات الرياضية المعقدة مثل حساب المثلثات والدوال والعمليات الإحصائية والهندسية .
4. إمكانية تحليل البيانات باستخدام الرسوم البيانيـــة Graphs ، مـــع إمكانيـــة التعديل على الرسوم وتلوينها وتكبيرها وتصغيرها ...
5. إمكانية استخدام البرنامج Excel كقاعدة بيانات (أي تقسيم البيانات ضمـــن سجلات وحقول لتسهيل عمليات الفرز والاستخراج المشروط والبحث) .
6. استخدام ميزة البرمجة والماكرو أي إمكانية تحويــــل العمليـــات الروتينيـــة الى عمليات تلقائية تنفذ كاملة بمجرد إنجاز خطوة واحدة .

بالإضافة إلى ميزات كثيرة أخرى يمكن التعرف عليها عند التعلم على استخدامه في مادة الحزم البرمجيات (1) .

(5–6–3) برامج إدارة قواعد البيانات

Database Management Program

في عصرنا الحالي أصبحت المعلومات وسرعة وصولها الى متخذ القرار أيـــاً كان موقعه عنصراً هاماً ومؤثراً على جوانب عديدة من المجتمع ، وذلـــك نتيجـــة التقدم العلمي والتطور التكنولوجي الذي رافقه فرزاً لحجم كبير من البيانات الأولية التي تحتاج الى نظم وأساليب للتعامل معها و معالجتها ، وقـــد لعـــب الحاســـوب بإمكانيته الهائلة في إدارة وتنظيم هذه البيانات وصنع المعلومات ، سميت هذه النظـــم

بـ نظم إدارة قواعد البيانـــات (Database Management Programs) واختصاراً (DBMS) . وسنقدم في هذه الفقرة شرحا للمفاهيم الأساسية لقواعـد البيانات :

(5-6-3-1) تعريف قواعد البيانات

يمكن تعريف قاعدة البيانات بأنها مجموعة من المعلومات والبيانـــات ذات صفات مشتركة والمخزنة بنظام بطريقة نموذجية تحقق نوعاً من الاستقلالية والمناعـة ضد التغيير عن البرامج التي نقوم باستخدامها حاضراً أو مستقبلاً إذا اضطررنا لتغيير أي منها ، وتتألف قاعدة البيانات من مجموعة من الســجلات مثـلاً ســجلات الموظفين أو سجلات الطلبة أو دليل الهاتف ، ســجلات المرضـى في مستشـفى سجلات المسافرين وحركة الطائرات في شركات الطيران ، سجلات حســابات العملاء في بنك ويدير قاعدة البيانات عادة نظام يدعى بنظام إدارة قواعد البيانـات. لذلك فان نظام إدارة قواعد البيانات يتكون من قسمين :

القسم الأول : قاعدة البيانات والتي هي مجموعة من البيانات المخزنة على شــكل ملفات مرتبطة مع بعضها البعض مثل حجوزات الطيران ، تســجيل الطلبة ، حسابات العملاء في بنك ...

القسم الثاني : برنامج ادارة قاعدة البيانات والتي هي عبارة عن مجموعة من البرامج التي تقوم بتشغيل ملفات البيانات الخزنة لتنفيذ العديد من العمليــات مثل الإضافة والحذف والتحديث والبحث والاسترجاع الجزئي والكلي والتبويب والفرز ...

(5-6-3-2) مكونات بيئة نظم قواعد البيانات

تتألف بيئة نظم قواعد البيانات من خمسة مكونات أساسية كما يلي :

1- المكونات المادية :Hard ware وتتألف من الحاسوب وتوابعه .

2- المكونات البرمجيه : Software وهي نظم خاصة بقواعد البيانات مثـــل

المترجمات ونظم التشغيل وبرمجيات الشبكات إذا كانت قواعد البيانات موزعة .

3- البيانات : Data وهي أهم مكونات نظم قواعد البيانات والتي من اجلها وجدت مثل هذه النظم .

4- الإجراءات : Procedures وهي التعليمات والقوانين التي تتحكم بقاعدة البيانات وكيفية الاستفادة منها والتحكم بالمخرجات .

5- المستخدمون : Users ويمكن تصنيف المستخدمون إلى أربعة مستويات:

1. مدير قواعد البيانات Database Administrator ، وهو شخص متخصص بقواعد البيانات و خبير لفترة طويلة في هذا المجال مهمته تحديد متطلبات قواعد البيانات من برمجيات وأجهزة والرقابة والضبط على أداء النظام ولديه الصلاحيات في التنسيق و التعامل مع الآخرين وحماية قاعدة البيانات من الاستخدام الخاطئ .

2. مصممو قاعدة البيانات Database Designer : ويكون إما شخص أو عدة أشخاص مهمتهم تحديد البيانات اللازم تخزينها ، تحديد افضل تراكيب هياكل البيانات الواجب استخدامها ، تصميم واجهات وشاشات التخاطب بين المستفيد و قاعدة البيانات ، تصميم الهيكل التنظيمي الداخلي لتخزين البيانات وأساليب الوصول إليها .

3. ميرمجو قواعد البيانات Programmers ، مهمتهم برمجة قواعد البيانات وجعلها قابلة للتنفيذ في تطبيق محدد .

4. مستخدمو قواعد البيانات Users ، وهم الأشخاص المستفيدون من تطبيقات أنظمة قواعد البيانات مثل : موظف التسجيل ، المحاسب ، الإداري ،

5-6-3-2) قواعد البيانات العلائقية Relational Databases

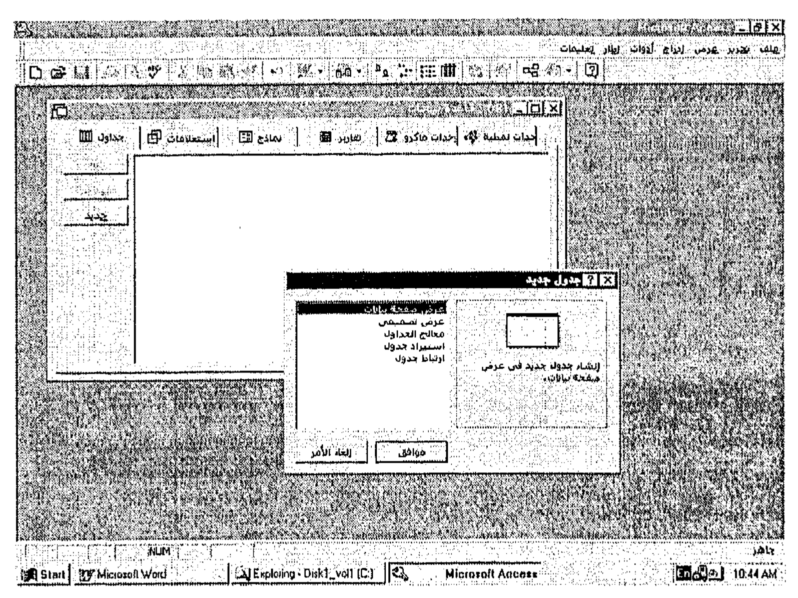
العلاقة هي بالأساس مصطلح رياضي ، وتمثل بجدول ذو بعدين مكوناً من صفوف وأعمدة ، تشكل هذه الجداول نوعاً من وسائل تخزين البيانات وحفظها في قاعدة بيانات وقد تكون الجداول ذات علاقة تبادلية فيما بينها ، لذلك سميت بقواعد البيانات العلائقية ويعتبر عام 1970 البداية النظرية لنموذج قواعد بيانات علائقية والذي اقترحه العالم كود E. F. Codd وفي مطلع 1980 انتشرت هذه الفكرة وأصبحت أداة لتنفيذ التطبيقات التجارية .

وهناك العديد من نظم إدارة قواعد البيانات التي تتفاوت في كفاءتها حسب سرعة استدعاء البيانات فيها بالإضافة إلى مدى سهولة استخدامها .

من أشهر هذه النظم برامج أسرة (DBASE) مثل (DBASE II, DBASE III , DBASE IV , FoxBASE FoxPro)، وبرنامج (Paradox) وبرنامج (Reflex) وبرنامج (Access) ...

ويعتبر برنامج آكسس أكثر برامج قواعد البيانات استخداماً وذلك لإمكانياته الضخمة ، الأمر الذي يسهل عملية إنشاء قاعدة البيانات وحفظها وتعديلها والبحث عن محتوياتها بأسرع وقت ممكن بالإضافة الى إمكانية ربط جداول قواعد البيانات مع بعضها البعض ، كما أنه توجد إمكانية استخدام لغة البرمجة (Access Basic)، وهذا بحد ذاته يدعم المبرمج ويوفر له إمكانيات برمجية ضخمة .

لا يمكننا هنا الخوض بتفاصيل الاكسس بل نتركه الى مادة حزم برمجيات (2) إن شاء الله للتعرف على جزئياته وكيفية العمل عليه و الشكل (13) يبين الشاشة الرئيسية للميكروسوفت اكسس Microsoft Access و أقسامها .



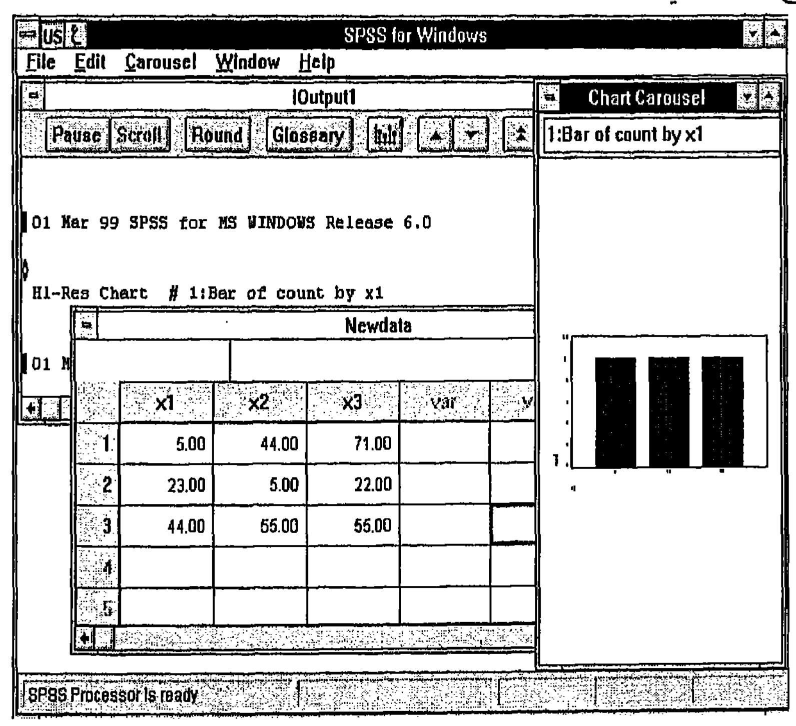
الشكل (13) الشاشة الرئيسية لمايكروسوفت اكسس Microsoft Access

(5-6-4) برامج الإحصاء المحوسب

مع التطور العلمي والتكنولوجي اصبح من الصعب إجـــراء التحليـــلات الإحصائية بالطرق اليدوية نتيجة لتضخم حجم البيانات الأولية المراد دراســـتها ، بالإضافة الى أن أغلب الدراســـات في مجـــالات العلـــوم الاقتصاديـــة والتربويـــة والاجتماعية والسياسية تعتمد وبشكل أساسي على تحليل البيانات إحصائياً مثـــل إيجاد الوسط الحسابي ، حساب معاملات الارتباط ومقـــاييس التشـــتت والترعـــة المركزية ... الخ من الخواص الإحصائية البسيطة والمعقدة .

ولهذا فقد دخل الحاسوب بإمكانياته الهائلة إلى هذا المجـــال العلمـــي الهـــام واصبح استخدامه في العلوم الإحصائية ضرورة حتمية ، فأصبح يقوم بتهيئة البيانات وتصنيفها وعرضها بيانيا و استخدامها في وضع النماذج الرياضية الإحصائيـــة وفي عمليات التنبؤ الاقتصادي والديموغرافي وإيجاد معادلات الانحدار والارتباط وجميـــع المعاملات الإحصائية ومتوسطاتها ، وكذلك وضع الدراسات الإحصائية في التحليل المالي والتحليل الإحصائي للعينات و الاختبارات الإحصائية والرياضية المختلفة ...

وهناك العديد من البرامج الإحصائية الخاصة التي تقوم ببعض أو كل ما ذكر سابقاً إلا أن أفضلها هي (SPSS) و (SAS) ، إلا أن برنامج الإحصاء المحوسب (SPSS) هو البرنامج الأكثر شيوعاً والأفضل استخداما في العلوم الإحصائية ، والذي سيدرس بالتفصيل في مادة الإحصاء المحوسب ، للتعرف على كيفية إدخال البيانات وحفظها وتحليلها واستخلاص النتائج . والشكل (14) يبين نافذة البرنامج الإحصائي SPPS .



الشكل (14) نافذة البرنامج الاحصائي SPSS

الفصل السادس

الشبكات الحاسوبية وتراسل المعلومات

Computer Network and Data Communication

(6 1) أهمية تراسل المعلومات:

استخدم الإنسان القديم، ومنذ العصور الأولى، عقله وذاكرته في التعلم واكتساب الخبرات، ومع مرور الزمن اكتشف أن واجبه وأهدافه لا تقف عند مرحلة التعلم وكسب المعرفة والخبرة، بل تعدى ذلك الى ضرورة نقل هذه الخبرات والمعارف الى الأجيال القادمة، عن طريق حفظ المعلومات على وسائل تخزين خارجية، وذلك لأن عقل الإنسان وذاكرته تنتهي بانتهاء حياته وقد تتأثر وتصاب بالخلل عند مرضه أو تعرضه الى أي ضغوطات أو تأثيرات نفسية أو أن تكون قدرته على حفظ المعلومات محدودة.

لقد تمثلت الذاكرة الخارجية قديما لحفظ المعلومات وتراسلها باستخدام الواح الطين ثم الأحجار والأخشاب مرورا بالجلد وغيرها من الوسائل البدائية الى ما نشاهده اليوم.

ومع تطور الإنسان ودخوله عصر النهضة الصناعية الحديثة، انتشرت الوثائق التي تحمل المعلومات في شتى المجالات وتنوعت الوسائط، وتفجر بركان هائل من البيانات والمعلومات التي ليست بحاجة الى تخزين فقط بل الى سهولة التعامل معها ومعالجتها وسهولة استرجاعها باسرع وقت ممكن واقل تكلفة، فأصبح الإنسان يستخدم لتسجيل وتراسل المعلومات الورق على شكل كتب ومجلات وخرائط، ثم

ابتكر أسلوب البريد لنقل معلومات سياسية أو حربية أو اجتماعيـــة أو انـــذارات، ومع مرور الزمن وتشعب الاعمال الحكومية والتجارية وتزايد حجـــم المعلومـــات صار التحدي الذي يواجه العلماء هو تحقيق الاتصال في الوقت المناسب، فتطـورت اساليب نقل المعلومة وظهرت أنظمة الاتصالات المتمثلة بالهاتف والبرق والفـــاكس والتلكس وأخيراً الحاسوب بإمكانياته الهائلة ووسائله التخزينيه المختلفة.

(6 – 2) الاتصالات والمعلوماتية:

منذ فجر الأربعينيات من القرن العشرين، بينما كان الحاسوب يظـــــهر الى الوجود، كانت أنظمة الاتصالات راسخة، وصناعة الاتصالات منتشـــرة في كـــل أنحاء المعمورة، ثم ما لبثت تكنولوجيا الحاسوب أن تطورت بسرعة هائلة من حيـث تخزين البيانات واسترجاعها ومعالجتها فاصبح الحاسوب والمعلوماتية يشكلان قـــوة يصعب فصلهما عن بعضها، وزاد من ترسيخ هذه القوة وأعطاها بعداً فريداً مــــن نوعه هو تطور الاتصالات وأجهزها ووسائلها. وظهور ما يسمى بالشبكات. فمـــا هو مفهوم الاتصالات الحديث..؟ وما هو مفهوم اتصالات البيانـــات..؟ وما هـــو مفهوم شبكة الاتصالات..؟ وعلاقة كل تلك المفاهيم بشبكة الانترنت..؟ هذا مـــا سنحاول ان نلقي الضوء عليه في هذا الفصل.

الاتصالات Telecommunications :

هي ارسال المعلومات بأي شكل مثل(صوت، بيانات، نصوص، صـــــور، اشكال) من مكان الى آخر باستخدام الوسائل الإلكترونية أو الضوئية.

اتصالات البيانات Data Communications :

هو مصطلح أكثر تخصصا، يصف عملية نقل واستلام البيانات من خـــلال خطوط الاتصال التي تربط بين حـــاسوب واحد أو اكثر ومعدات ادخال واخـراج مختلفة.

شبكة الاتصالات Telecommunications Network:

هي أي ترتيب يقوم فيه المرسل بارسال رسالة الى مستلم عبر قنوات اتصال من نوع ما، والشبكة تحتوي على حواسيب مرتبطة ببعضها البعض وببعض وحدات طرفية تقوم بمعالجة وتبادل البيانات و المعلومات.

(6 3) الشبكة: (تعريفها أنواعها أهميتها)

الشبكة، من وجهة نظر المعلوماتية، هي مجموعة مواد ومعدات معلوماتية متصلة مع بعضها البعض، بكلام آخر: الشبكة هي عبارة عن مجموعة من الحواسيب Computers أو حاسوب كبير رئيسي ترتبط به مجموعة من طرفيات Terminals أو حواسيب صغيرة، ترتبط مع بعضها البعض بخطوط الاتصال(السلكية او اللاسلكية)، بحيث يمكن لمستخدميها المشاركة في الموارد المتاحة(من معلومات وبرامج) ونقل وتبادل المعلومات فيما بينهم.

فإذا كان الربط والاتصال بين الحواسيب بوسائط سلكية وفي منطقة محددة فعندئذ تسمى شبكة محلية LAN (Local Area Network)، أما إذا كان الربط عن طريق خطوط الهاتف أي باستخدام نفس مسار المكالمات الهاتفية في تبادل المعلومات الحاسوبية تسمى بالشبكة الواسعة WAN (Wide Area Network) وغالبا ما تكون ممتدة هذه الشبكة على مساحة جغرافية كبيرة وتستخدم عادة بين البنوك ومراكز البيع الواسعة وفي نظام الانترنت، وسنتناول في الفقرة التالية مكونات أنظمة الشبكات ومعداتها المختلفة بالتفصيل.

إن أهمية الشبكات تأتي في أنها ساهمت وتساهم في رفع كفاءة التشغيل ودعم اتخاذ القرار في شتى مجالات الحياة من خلال:

1. كفاءة وسرعة الاتصال وسهولة نقل وتبادل المعلومات.
2. التشغيل الاقتصادي للأجهزة وذلك بالمشاركة في استخدامها.
3. المشاركة في البرمجيات.

4. المشاركة بالمعلومات المتوفرة وقواعد البيانات.

5. تطبيق مبدأ المعالجة الموزعة(Distributed Processing) التي تعني توزيع المهام على عناصر الشبكة مما يؤدي الى سرعة انجازها ورفع اقتصاديات تشغيل هذه العناصر.

(6 4) مكونات انظمة الشبكات:

يمكن تقسيم العناصر المكونة لأنظمة الشبكات الحاسوبية من:

1. معدات الشبكات Network Hardware
2. برمجيات الشبكات Network Software
3. قنوات الاتصال Communication Channels

وسنتكلم عن كل من هذه المكونات بالتفصيل.

(6 4 1) معدات الشبكات: Network Hardware:

وتشتمل معدات الشبكات على الطرفيات ووحدات التحكم الرقمية ووحدات المواجهة البينية وأجهزة الحواسيب المختلفة:

أولا: الطرفيات Terminals:

إن أي وحدة ادخال وإخراج مثل الطابعات-الراسمات-الأجهزة الصوتية -قارئ البطاقات المثقبة ، تدعى طرفيات، وذلك لوجودها في نهاية خطوط الشبكة، بالإضافة الى ذلك فانه يطلق مصطلح الطرفيات على الحواسيب الصغيرة التي تكون في نهاية خطوط الشبكة وذلك لان الحاسوب يمكنه إرسال البيانات واستقبالها من حاسوب آخر شبيه به أو اكبر منه.

ثانيا: وحدات التحكم الرقمية Modems:

وهي عبارة عن جهاز يحول البيانات والمعلومات، من معلومات تناظرية(Analog) تسلك مسار المكالمات الهاتفية، الى معلومات رقمية(Digital) لكي تناسب الحاسوب لأنه كما نعلم ان الحاسوب والطرفيات تستخدم اشارات

عددية(Digital Signals) ذات جهود منفصلة تمثل بواسطة 1,0 بينما خطوط الهاتف تستخدم اشارات تناظرية(Analog Signals) ذات جهود متصلة، ولذلك قبل نقل الاشارات العددية عبر خطوط الهاتف يجب تحويلها الى اشارات تناظرية وتسمى هذه العملية بتعديل الاشارة Modulation وفي الطرف الآخر، عند استقبال البيانات يجب تحويل الاشارات التناظرية الى اشارات عددية يستطيع الحاسوب فهمها ومعالجتها، وتدعى هذه العملية بإعادة التعديل Demodulation ولذلك فإن وحدة التحكم الرقمية تستخدم للتعديل وفك التعديل.

ثالثا: وحدات المواجهة البينية Interface Units:

وظيفة وحدات الواجهة البينية توجيه عمليات ارسال واستقبال البيانات مما تزيد من كفاءة النظام بشكل كبير، ومن هذه الوحدات:-

1. جهاز مضاعفة الإرسال: Multiplexer: عبارة عن جهاز يقوم بدمج الاشارة القادمة من عدة طرفيات في قناة واحدة بشكل متزامن، وبالتالي يمكن أن تشارك عدة طرفيات بنفس قناة الاتصال، اذ يقوم هذا الجهاز بتقسيم خط الاتصال ذو السرعة العالية الى مجموعة من القنوات ذات سرعة أقل.

عادة تتم مضاعفة الارسال من خلال اسلوبين هما:

أ- أسلوب المضاعفة الموجية: حيث يتم تقسيم قناة الاتصال الى عدة قنوات من خلال تقسيم الحزمة الموجية الى حزم موجية أصغر تستخدم كل منها كقناة اتصال مستقلة.

ب- أسلوب المضاعفة الزمنية: حيث يتم تخصيص قناة عامة لخدمة عدة قنوات بحيث يكون لكل من هذه القنوات وقت محدد يستخدم خلالها القناة العامة لتمرير بياناتها.

2. المجمعات أو المركزات Concentrators: وهي أجهزة تحتوي على معالجات متوسطة(Micro Processors)، ويمكن أن تكون جهاز حاسوب صغير أو متوسط، وظيفتها دمج وتجميع البيانات الواردة اليها من عدة طرفيات وتخزينها بشكل مؤقت ثم ارسالها الى جهتها المطلوبة عبر خطوط اتصالات سريعة.

3. مفاتيح الرسائل: Message Switchers: وهو عبارة عن جهاز يستقبل جميع الرسائل من الطرفيات المختلفة، ويقوم بتحليل الرسائل ومعرفة هدفها ومسارها الصحيح ثم توجيهها الى الهدف أو الموقع المطلوب. نذكر هنا أنه هناك مفاتيح رسائل تمتلك القدرة على تخزين الرسائل اذا لم يكن خط النقل المطلوب متوفرا لحظة وصول الرسائل مما يخفف ذلك على الحاسوب الرئيسي مهمة توصيل الرسائل ويزيد من كفاءة الشبكة.

رابعا: الحواسيب Computers

تختلف الحواسيب في الشبكة من حيث الحجم والقدرات والإمكانيات، وعادة في الشبكات الكبيرة يكون الحاسوب الذي يمثل مركز المعالجة الرئيسي من النوع الكبير إما في الشبكات الصغيرة فيكون من النوع المصغر.

ويدعى الحاسوب الذي يمثل مركز المعالجة الرئيسي بالحاسوب المضيف Host Computer أي الحاسوب الذي يستعمل الخدمات المقدمة من الحواسيب الصغيرة التي بدورها تدعى الحواسيب الملقمة.

(6 4 2) برمجيات الشبكات: Network Software

وتتكون برمجيات الشبكات من برمجيات التحكم في الاتصالات بين كل من الحاسوب المضيف والحواسيب الطرفية بالإضافة الى برمجيات المراقبة وتوجيه البيانات والتحكم في تدفقها عبر خطوط الشبكة.

وتوفر حزم البرمجيات المستخدمة في الشبكات مجموعة من الوظائف الهامة منها:

1. رقابة فتح واغلاق خطوط النقل في الوقت المناسب.
2. تحديد أخطاء نقل المعلومات وتصميمها او اعادة نقلها عند حدوث خطأ.
3. تحديد مسار نقل البيانات وترتيب عمليات النقل بتسلسل معين حسب أولويات محددة.
4. حماية الشبكة من الاستخدام الخاطئ غير المسموح به.

(6-4-3) قنوات الاتصال Communication Channels

وهي الممرات التي يتم فيها تراسل المعلومات والبيانات، وتسمى أيضا بخطوط البيانات Data Links أو خطوط الاتصال Communication Links وتربط هذه الخطوط مختلف الحواسيب والطرفيات والأجهزة الأخرى في الشبكة مع بعضها البعض. ولهذه القنوات أنواع مختلفة نصنفها ضمن مجموعات حسب صفات معينة وهي:

أولا: حسب تركيبها الفيزيائي:

وتقسم إلى ثلاثة أنواع وهي:

1. الكوابل المعدنية Wire Cables

وهي التي تستخدم في الخطوط الهاتفية، هذه الكوابل تحتوي على حزم من اسلاك النحاس المجدولة(Twisted Pairs)، تحمل المعلومات على شكل إشارات تناظرية، وتتميز بكلفتها المخفضة و توفرها في جميع انحاء العالم وسهولة استخدامها ولكن يعاب عليها بحساسيتها للتشويش الإلكترومغناطيسي مما يؤدي الى الحد من سرعة نقل البيانات عبرها. لذلك تم اختراع كوابل جديدة للتغلب على هذه المشكلة تدعى الكوابل المحورية Coaxial Cables وتتكون من كابل مركزي يحيط به حزمة اسلاك خارجية كدرع واق تحمي الكابل المركزي وتقلل من التشويش الالكترومغناطيسي، وتمتاز أيضا بسرعتها في نقل المعلومات وامكانية استخدامها تحت الارض وتحت الماء.

2. الامواج المصغرة Micro Waves

وهي نفس أمواج الراديو بتردد عال، وتستخدم لنقل البيانات مع استخدام محطات تقوية تلتقط هذه الموجات ثم تعيد بثها بعد تقويتها، مما يسمح بنقلها الى مسافات بعيدة، ويمكن استخدام هذه الموجات أيضا في اتصالات الأقمار الصناعية Communications Satellite التي تلتقط الأمواج المصغرة من احدى محطات الأرض وترسلها الى محطات أخرى.

3. كوابل الألياف الضوئية Fiber Optic Cables

وهذا النوع يتكون من حزمة من الموصلات الزجاجية المصنوعة من السيلكون النقي والقادرة على نقل الضوء على شكل نبضات فعالة وغير فعالة(On and Off pulses) تولدها أشعة الليزر.

ثانيا: حسب كمية البيانات التي يمكن ارسالها عبر هذه القناة في وقت معين.

وهذا يعتمد على عرض حزام القناة(مدى الترددات التي يمكن نقلها)، ويمكن تصنيف القنوات حسب عرض الحزام الى:

الحزام الضيق Narrow Band	: أقل من 2000 هرتز ونسبة النقل فيها تتراوح من 7 الى 30 رمز / ثانية.
الحزام الصوتي Voice Band	: 3000 هرتز ونسبة النقل فيها حوالي 960 رمز / ثانية.
الحزام العريض Board Band	: اكثر من 3000 هرتز ونسبة النقل فيها تصل إلى 40 مليون رمز / ثانية، وكوابل الامواج الألياف الضوئية والأمواج المصغرة هي قنوات الحزام العريض.

ثالثا: حسب اتجاه نقل البيانات: وتقسم الى:

القنوات البسيطة Simplex Channels: والتي يتم فيها نقل البيانـــات باتجـــاه واحد فقط ولنقل البيانات باتجاهين يجب استخدام قناتين مســتقلتين من هذا النوع واحدة للارسال والأخرى للاستقبال.

القنوات نصف المزدوج Half-Duplex Channels: وتقوم أيضا بنقل البيانـات باتجاه واحد فقط ولكن يمكن تغيير اتجاه عملية النقل، أي أن النقــل يتم أولا باتجاه واحد ثم يتم عكس الاتجاه ونقل البيانات في الاتجـــاه المعاكس.

القنوات الكامل الازدواجية Full-duplex Channels: تتم عملية النقل في كلا الاتجاهين في الوقت نفسه، مثل خطوط الهاتف.

رابعا: حسب أسلوب نقل البيانات: وهناك أسلوبان:

النقل المتزامن Asynchronous: وهذا يتطلب ارسال رمز واحـــــد في الوقــت الواحد،ولفصل الرموز عن بعضها يضاف بت البدايـة(Start bit) و بت النهاية(End bit) الى طرفي شيفرة الرمز المكونة من(8bit).

النقل المتزامن synchronous:ويتضمن نقل كتلة من الرموز عشـــوائيا، وهــذه الكتلة من الرموز ترسل بدون بت البداية والنهاية، ويتــم توقيــت عملية النقل بحذر شديد.

خامسا: حسب طريقة ربط القنوات التي تربط الطرفيات بالحاسوب:

قنوات الاتصال المباشر Point-To-Point Channels: أي تربط كل طرفية مع الحاسوب عبر قناة منفصلة، وهذه الطريقة غاليــــة الثمــن، وتكون هذه الطريقة مجدية اقتصاديا عندما تكون كمية

البيانات المنقولة كبيراً جداً او في الحـــالات الـــتي تتطلـــب استجابة سريعة.

قنوات الاتصال المتعدد Multidrop Channels: تربط عـــدة طرفيـــات مـــع الحاسوب عبر قناة واحدة، وهذه الطريقة اقتصادية إذ تشترك عدة طرفيات بقناة واحدة ووحدة واجهة بينية واحدة، نشير هنا الى أنه في هذا الاسلوب من الربط يمكن لعدة طرفيـــات استلام البيانات بالشكل المتزامن أما لارسال البيانات فـــلا يستطيع الا طرفية واحدة فقط.

(6 5) شبكات المعالجة الموزعة

Distributed Data Processing Networks

بشكل عام يوجد نوعان من الشبكات هما:

1. شبكات الاتصـــالات Communication Networks: وظيفتـــها نقـــل البيانات وهذه تستخدم الحواسيب للمساعدة في عمليات النقل.
2. شبكات المعالجة الموزعة Distributed Processing Networks: وهـــي الشبكات التي تربط بين مكونات مادية مختلفة بهدف المشـــاركة في المـــوارد وامكانيات المعالجة، أي يكون لدينا عدة حواسيب كل منها يقوم بعمليـــات المعالجة الخاصة به بدلاً من حاسوب واحد يتحمل جميع اعباء المعالجة، ويمكـــن أن تحتوي الشبكة على حاسوب مركزي يقوم بعمليات المعالجة المشتركة ممـــا يؤدي الى تفرغ كل حاسوب لاتمام عملية المعالجة الخاصة به.

(6 6) هيكلية الشبكات Networks Topology

المقصود بهيكلية الشبكات هو نوع وشكل الاتصال بين المكونات الماديـــة للشبكة ومن اهم واشهر هذه الاشكال:

1. الاتصال النجمي Star Topology(الشبكة النجمية)

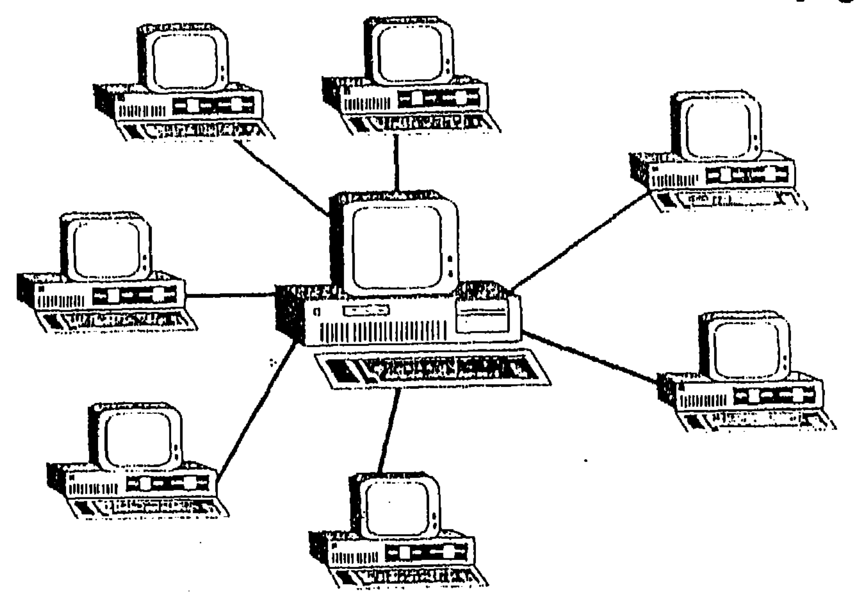
2. الاتصال الحلقي Ring Topology(الشبكة الحلقية)

3. الاتصال الخطي Bus Topology(الشبكة الخطية)

4. الاتصال الهرمي Hierarchical Topology (الشبكة الهرمية)

(6-6-1) الشبكة النجمة Star Network:

وهي ابسط وأقدم أشكال الشبكات، وهي تتكون من حاسوب مركـــزي يسمى الخادم Server يتصل به مجموعة من الحواسيب الشخصية أو الطرفيـــات كما في الشكل(1)، حيث يعمل الحاسوب Server كنظام تحكم يتم من خلالـــه السيطرة على كافة الاتصالات بين الاجهزة المتصلة به، وأي انتقال للمعلومات يتـــم عن طريقه.



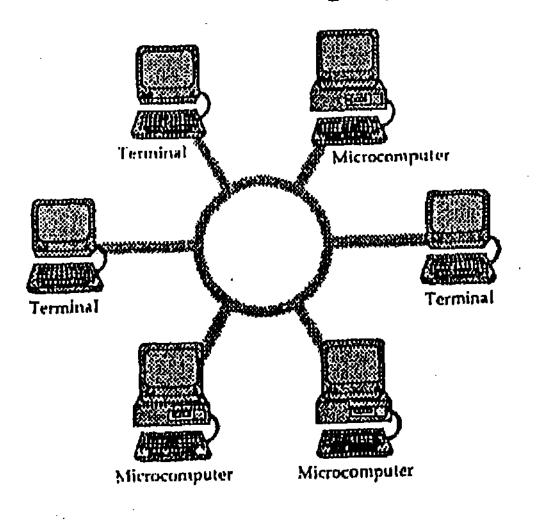
الشكل(1) نموذج الشبكة النجمية

ومن عيوب هذه الشبكة انخفاض درجة الموثوقية بها بسبب أن تعطل الجـــهاز المركزي يؤدي الى تعطل الشبكة باكملها، اضف الى ذلك أنه هناك ضياع للوقـــت بسبب زيادة وقت الانتظار بالنسبة للحواسيب الطرفية وذلك بسبب عدم امكانيـــة انجاز اكثر من اتصال واحد في الوقت نفسه.

(2-6-6) شبكة الحلقة Ring Network

وهذه الشبكة تتكون من عدة حواسيب كل منها يتصل بالآخر مباشرة، وتتصف هذه الشبكة بعدم الحاجة لوجود حاسوب مركزي، لذلك فالحواسيب تأخذ شكل دائرة او حلقة كما في الشكل(2)، وهذه الشبكات تستخدم في المنشآت التي لا تحتاج الى تحكم مركزي لفروعها الموزعة، ويتم تنسيق الاتصالات خلال هذه الشبكة باستخدام بروتوكول خاص، والبروتوكول هو عبارة عن (مجموعة من القواعد والقوانين والاجراءات تحدد كيفية عمل الشبكة ووظائفها)، يسمى بروتوكول علامة المرور Token-Passing Protocol، هذه العلامة هي عبارة عن اشارة تحكم يتم بواسطتها تحديد الجهاز المسموح له بنقل المعلومات عبر الشبكة، وبالتالي حسب هذا المبدأ فان جهاز واحد فقط يستطيع استخدام الشبكة في الوقت نفسه.

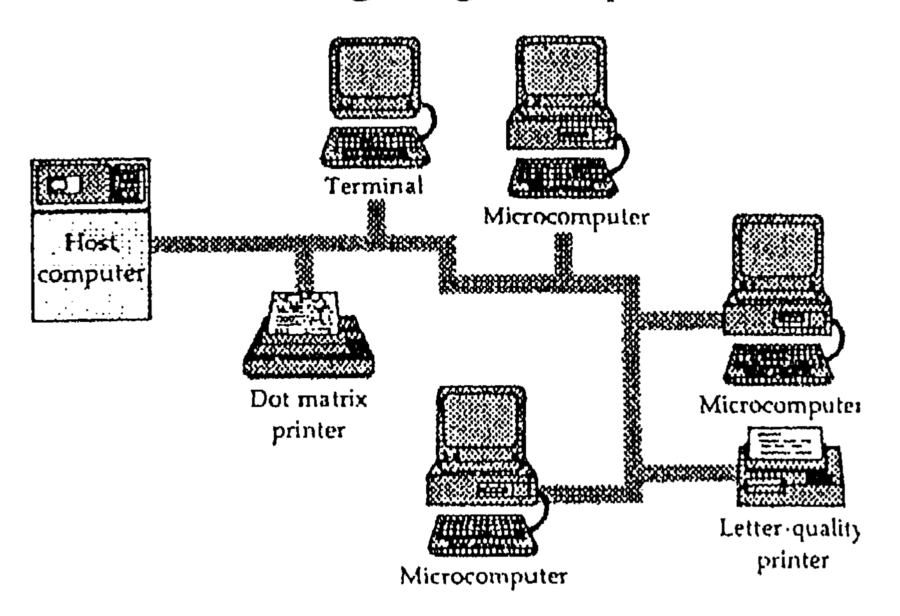
لذلك هذه الشبكة بدرجة عالية من الموثوقية، وذلك لأن تعطل أحد الأجهزة في الحلقة لا يؤثر على عمل باقي الاجهزة، بالاضافة الى قابليتها للتوسع واضافة مكونات اخرى. والشكل (2) يمثل نموذجا للشبكة الحلقية.



الشكل (2) نموذج لشبكة الحلقة

(3-6-6) الشبكة الخطية Bus Network

في هذا النوع من الشبكات ترتبط الاجهزة ومحطات التشغيل والطرفيات بواسطة كابل اتصال خطي Bus مفتوح ومزدوج الاتجاه، كما في الشكل (3).



الشكل (3) نموذج للشبكة الخطية

وللتحكم في مرور المعلومات خلال هذه الشبكة، فانه يستخدم بروتوكولات خاصة تسمى Carrier-Served Access اكتشاف التصادم و اختصارا (CASM) ومن هذه البروتوكولات:

- بروتوكول مانع التصادم (Collision Avoidance)(CSMAICA) والذي يتضمن قيام جهاز واحد بنقل المعلومات عبر الشبكة في الوقت نفسه.
- بروتوكول اكتشاف التصادم(Collision Detection (CSMA/CA، لتنظيم عملية النقل حسب الاسبقية في طلب خدمة النقل لمنع التشويش.

ميزات هذه الشبكة:

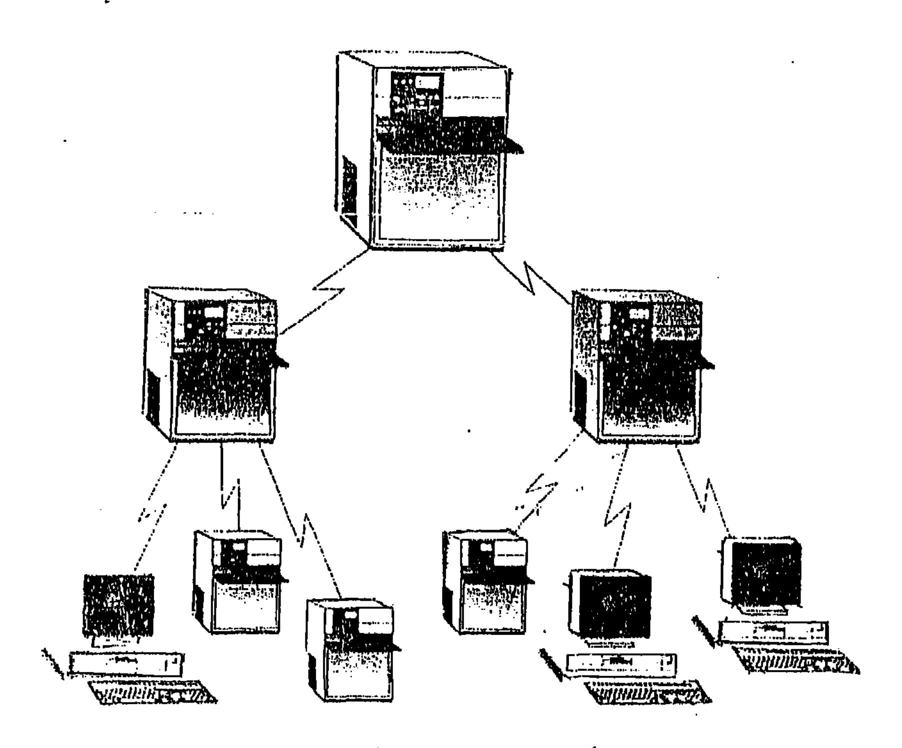
1. البساطة مع المرونة والقابلية للتوسع.

2. توفير البروتوكولات المنظمة لعمل الشبكة.

3. درجة الموثوقية عالية فيها حيث أن الشبكة لا تتوقف عن العمل الا بتوقف إحدى محطاتها.

(6-6-4) الشبكة الهرمية Hierarchical Network

تتصل الحواسيب فيها على شكل هرم حسب مستوى أهميتها، والشكل (4) يسمى الحاسوب في قمة الهرم بالحاسوب الجذر(Computer Root) وهو يتحكم في الحواسيب التابعة له، وهي بدورها تتحكم بالحواسيب المتفرعة منها وهكذا، ولذلك تدعى أحيانا باسم الشبكات السيد/الخادم (Master/Slave Network) وعادة يستخدم هذا الاسلوب في المنشآت التي تعتمد ادارتها على سلسلة من الأوامر والتعليمات وكما في الشكل (4).



الشكل (4) نموذج لشبكة هرمية

6-7) الانترنت Internet

6-7-1) ماهية الانترنت

الانترنت هي شبكة معلوماتية عالمية مؤلفة من مجموعة كبيرة من شبكات الحاسوب موصولة مع بعضها البعض، لقد كانت البدايات الأولى لهذه الشبكة مقتصرة على عدة اجهزة حواسيب، لا يزيد عن عشرة حواسيب مضيفة، ثم ما لبثت ان تطورت وتوسعت واصبح بإمكان أي شخص من خلال حاسوبه الشخصي في المكتب أو البيت أو في أي مكان قابل للربط مع هذه الشبكة الحصول على المعلومات التي يحتاجها من مصادرها وحتى وان كانت هذه المصادرة في أقصى بقاع العالم شريطة أن تكون هذه المصادر مرتبطة بهذه الشبكة العالمية، والتي ترتبط بدورها بملايين الحواسيب ومراكز المعلومات المنتشرة في العالم لتشكل ما يعرف بالنسيج العالمي الواسع للمعلومات او شبكة المعلومات العالمية World Wide Web (WWW).

لقد كانت التجربة الأولى عام 1969م، عندما تم توصيل حواسيب وزارة الدفاع الامريكية، وبدأ كل منهم يتحدث مع الآخر عبر هذه الشبكة الصغيرة، وبعد نجاح هذه التجربة تم توصيل حاسبات هذه الشبكة مع جهات اخرى، وحققت أيضا نجاحا باهرا وأخذ عدد الأجهزة الحاسوبية المتصلة مع هذه الشبكة تزداد، ثم نشأت مجموعة من الشبكات، واصبحت شبكة الانترنت تجمع مجموعة من شبكات المعلوماتية المختلفة، وهنا نشير الى ان شبكة الانترنت ليست هي الوحيدة في العالم بل هناك شبكات عديدة تهتم بالمعلوماتية منها شبكة كمبيوسيرف(CompuServe) وشبكة أمريكا أون لاين (America on Line) وشكبة بت نت (Bit Net) وهذه الشبكات تملكها شركات مقرها الولايات المتحدة الامريكية، مسؤولة عن تعديل المعلومات فيها بشكل مستمر والتحكم بنوعية خدماتها ومعلوماتها.

أما شبكة الانترنت فوضعها مختلف تماما، اذ ان هذه الشبكة لاتملكـــها أي جهة معينة في العالم، ولا سلطان عليها من أي شخص، ولا هيمنـــة أي منظمـــة دولية، بل انها ملك للجميع، يستطيع أي شخص الاتصال بها ويخصص له عنـــوان ويصبح جزءا منها وشريكا فيها.

يعود انتشار هذه الشبكة و توسعها لعدة أسباب أهمها:

1- الوصول المجاني الى المعلومات، الا في حـــالات قليلـــة، حسب أهميـــة المعلومات، وما تدره من أرباح، أما بخصوص الأمـــوال الـــتي تدفـــع الى الجهات الموزعة للخدمة عند الاشتراك في خدمة الانترنت فـــهي لصيانـــة الخطوط وزيادة سرعتها وبث البرامج الموصلة للخدمة.

2- الزيادة المستمرة في عدد المشتركين، وزيادة مـــوارد الخدمـــات والـــدول المشتركة على الشبكة، مما يزيد من كميات المعلومات المتاحة.

3- بناء الشبكة على أساس نظام(العميل/الخـــادم)(Client/Server) هـــذا النظام في خدمات الانترنت يعطي إمكانية توصيل جميـــع الحواســـيب الى الشبكة، بحيث يسمح لها بإمداد خدمات واستلام خدمات مـــن بعضـــها البعض.

ملاحظة:

* العميل Client: هو الذي يتعامل مع المســـتخدم، ومـــن مســـؤولياته قبـــول مدخلات لوحة المفاتيح واظهار البيانات على الشاشة للمستخدم.

* الخادم Server: مسؤول عن انجاز جميع العمليات الحاسوبية بواسطة العميل مثل استرجاع البيانات بالنيابة عن العميل وانجاز العمليـــات الحاســـوبية، وإرســـال البيانات الى العميل.

* نظام(العميل/الخادم) يمكن تشغيله على حاسوب واحد، الا انه في الغالب يتـــم تشغيله على حاسوبين مختلفين، احدهما عميل والاخر خادم.

(2-7-6) لمحة تاريخية عن شبكة الانترنت:

سنعرض في هذه الفقرة تطور شبكة الانترنت منذ بدايتها وحتى الآن، وأهم شبكات المعلومات المتصلة معها وما تقدمه من خدمات:

1. **ما قبل تاريخ الشبكة**: حيث أنشأت وزارة الدفاع الامريكية DOD(Department Of Defense) عام 1957 وكالة لمشاريع الأبحاث المتقدمة(ARPA)Advanced Research Project Agency تهتم بتطوير العلوم التي تخدم الاحتياجات العسكرية. ويشار الى ان تاريخ الانترنت بدا في الولايات المتحدة في الستينات وكانت عبارة عن شبكة تعتمد طريقة تبديل الحزم وهي تقنية نقل معلومات ترتكز على تقسيم الرسالة الى سلسلة من الحزم التي ترسل عبر الشبكة.
2. **أربانت ARPA net**: جرى أول تحقيق عملي لشبكة أربانت في جامعة كاليفورنيا في لوس انجلوس 1969 و تألفت الشبكة من أربعة حواسيب، ثم أضيفت عقد أخرى الى الشبكة وبشكل تدريجي وخاصة عقد من معهد بحوث ستانفورد في جامعة كاليفورنيا في سانتار باربارا وكذلك عقد من جامعة يوتا، حتى اصبح عدد العقد 62 موقعا عام 1974 و عام 1981 اكثر من 200 موقع.

إنترنت Internet: جرى عرض أول تحقيق تجريبي عملي يتضمن 40 عقدة لشبكة أربانت خلال المؤتمر العالمي الأول حول الاتصالات الحاسوبية في واشنطن عام 1972 وتركزت المناقشات حول توحيد بروتوكولات الاتصالات، و نتيجة لهذا المؤتمر تم وضع المخطط الأولي لبنية الشبكات العالمية المتحدة في المستقبل حيث تتمتع الشبكات الجزئية بإستقلالية كبيرة وتتصل فيما بينها. وفي عام 1979 تم انشاء مجلس للتشكيل والتحكم بالانترنت هدفه مراقبة تطور الشبكة ويسمى:

Internet Configuration Control Board (ICCB)

4. ميلنت **MIL net**: قسمت شبكة اربانت عام 1983 الى شبكتين اربانت وميلنت، ارتبطت شبكة ميلنت بشبكة المعطيات الدفاعية، أما اربانت فقد كانت العمود الفقري لشبكة إنترنت عام 1990 ومن ثم ضمت اربانت الى شبكة القاعدة العلمية الوطنية National Science Foundation Net (NSF net) التي أضحت بدورها العمود الفقري لشبكة إنترنت بين عامي 1990 1995، وفي عام 1995 بدلت هذه الشبكة بمجموعة شبكات كبيرة متصلة مثل ANS net , Sprint net , MCI net

5. يوزنت **USE net**: هي ليست شبكة بل خدمة مؤتمرات الكترونية تستخدم الشبكة كحامل للمعلومات. نشأت يوزنت عام 1979 وكانت على شكل ثلاث حواسيب هدفها نقل الاخبار بين مستخدمي نظام يونيكس، واستخدم بروتوكول UUCP(Unix To Unix Copy): وهو بروتوكول لتبادل الرسائل والملفات إلكترونيا بين مستخدمي نظام Unix، سرعان ما ازداد عدد مواقع اربانت المستثمرة لأخبار يوزنت الأمر الذي أدى الى استبدال بروتوكول UUCP بروتوكول نقل اخبار الشبكة

(NNTP) : (Net News Transfer Protocol).

6. بت-نت **Bit net**: أنشأت في جامعة نيويورك لتأمين نظاما للمؤتمرات الالكترونية ولكن سرعان ما دمجت مع شبكة CS net عام 1989 لتشكيل شبكة التعاون من أجل شبكة البحوث والتعليم

Corporation for Research and Education Network (CREN)

7. **شبكات الخدمة الآنية**: تكون هذه الشبكات متصلة مع شبكة إنترنت وتقدم خدمات آنية لمستخدميها على المستوى القومي والعالمي وتوجد بين هذه الشبكات وشبكة الانترنت عبارات للرسائل، ومنذ فترة وجيزة أمنت معظم هذه الشبكات لمستخدميها ولوجا الى شبكة الانترنت الى حد ما. أهم هذه الشبكات:

أ. كمبوسرف **CompuServe** : تأسست عام 1979 تؤمن هذه الشبكة خدمات البريد الالكتروني والمناقشات على الشبكة ونقل الملفات ويتم الوصل مع الشبكة بواسطة مودم.

ب. امريكا أونلاين(**America Online(AOL** :

تعتبر إحدى أكبر الشبكات الامريكية وخاصة بعد شرائهما لعدة شركات أخرى مثل: ANS net في شباط 1995، التي تعتبر إحدى شبكات إنترنت الأساسية في امريكا، و تهتم شركة AOL بالأمور الإعلامية.

ج. دلفي إنترنت **Delphi Internet**:تهتم بالمعلومات الإعلامية منذ عام 1995 اذ تتضمن عددا كبيرا من ا لصحف العالمية الهامة مثل(Sun, New) ومحطات تلفزيونية(Fox الشبكة الامريكية الرابعة، BSKYB في انكلترا) وأقمار صناعية وشركات سينمائية

د. بروديجاي **Prodigy**: تعتبر هذه الشبكة فرع مستقل من IBM وSears، أنشأت عام 1988.

هـ **Internet MCI**: التي وفرت منذ زمن طويل خدمة البريد الالكتروني(MCI Mail).

و. شبكة مايكروسوفت **Microsoft Network**: تؤمن خدمة البريد الالكتروني وخدمة نقل الملفات وخدمة المناقشات وعدد كبير من الخدمات التجارية لكل حاسوب يعمل تحت ويندوز 95 ومرتبط مع مودم أو مع ويندوز 98.

ز. **Apple Work**: شبكة خدمة آنية لشركة Apple.

ح. ويل **Well**: تأسست هذه الشركة لتأمين الولوج الى الانترنت.

(6-7-3) خدمات الانترنت

خدمات الانترنت كثيرة جدا ولذا سوف نقتصر على أهم هذه الخدمات:

1. خدمات الاتصالات:

أ. البريد الالكتروني(Electronic Mail) وفي هذه الخدمة يمكن استقبال وارسال الرسائل من و الى أي شخص في العالم يستخدم هذه الشبكة ان أهم البرامج المستخدمة في البريد الالكتروني هي:

-ايدوارا(Eudora)

-تيلنت(Telnet)

للدخول الى برنامج(Eudora) يتم بالشكل التالي:

(com />.(Eudora , < http://www.qualcom

ب. مجموعات النقاش: تعود شبكات الاخبار(News) على الشركات بفائدة كبيرة، فهي تحوي على معلومات متعددة،كما أنها تسمح بتأمين وصل مباشر بين الشركات وزبائنها، مما يؤدي الى تحسين الخدمات التي تقدمها الشركة، اما الافراد فبإمكانهم استخدام الأخبار News للتسلية أو لإجراء مقابلات او اعلانات صغيرة

(Fr.Petites announces. Vehicules) News أو استدراج عروض News : offers. Fr. Jobs.

ج. خدمات حوار إنترنت : تسمح Internet Relay Char (IRC) لمستخدمي إنترنت بالتخاطب في الزمن الحقيقي بشكل نصي، يمكن الاستفادة منها في التطبيقات المهنية او الخاصة، اما Internet Phone فتسمح باجراء مخابرات هاتفية باستخدام شبكة إنترنت.

2. خدمات الدعاية والاعلان

أ. الخدمات المجانية: معظم المعلومات المتوفرة على الشبكة تقدمها الهيئات التي تنتجها مجاناً. وتقدم معظم الشركات المعلومات خاصة بمنتجاتها

وكمثال على ذلك فان شركة مايكروسوفت تقدم للجميع تصميمـــــات برمجية Patches وبرمجيات مجانية ومعلومـــات فنيـــة عـــن منتجاتهـــا، ومعلومات حول الشركة ذاتها مثل (التقرير السنوي، اعلانات الصحافـــة، عروض العمل، ... الخ).

كذلك تنشر الجامعات والمدارس ومراكز البحـــوث كمية من المعلومـــات ان معظم المكتبات الضخمة تتواجد أيضا على الانترنت، كما تنشر وكالة الاستخبارات المركزية CIA كتابها عن الجرائم على الانترنت.

<http://www.odci.gov/cia/publications/pubs.html>

كما تضع وكالة الفضاء الــ NASA أخبار مهماتها الســـابقة والحاليـــة واللاحقة وتنشر معلومات خاصة بالمكوك الفضائي.

<http://shttle.nasa.gov/>

كما تعرض المجموعة الاوربية وثائق اقتصادية وسياسية

<http://www.echo.lu/>

ب. الخدمات التجارية: تقدم بعض المؤسسات او شركات الأنباء المحلية عـــبر شبكة الوب أو عبر News مثل الاخبار المالية وأسعار البورصة ونســـب صرف أسعار العملات ويمكن ان تكون شبكة الانترنت وســـيلة إعـــلام جديدة قادرة على مناقشة أو تكميل وسائل الاعلام التقليدية كالصحافـــة والمذياع أو التلفاز.

ج. النشر الآلي: هناك عدد من المجلات التي تنشر إلكترونيا مثل عالم إنـــترنت Internet World على العنوان:

<http://www.mecklesweb.com:80/mags/iw/iwhome.htm>

كما ان معظم الصحف الامريكية موجودة على العنوان:

<http://www.pathfinder.com>

د. بنوك المعطيات: يمكن الوصول بسهولة الى العديد من بنوك المعطيات عـــن طريق الانترنت.

3. **الخدمات المساعدة:** تصنف ضمن هذه الفئة كافة الخدمات عن بعـــد والـــتي تهدف الى بيع منتج ما او خدمة عن طريق الشبكة، حيث نجد اليـــوم علـــى الشبكة كل شيء من ازهار مقصوصة الى بياضات، مرورا بالكتب والبرمجيـات والسيارات ...الخ.

4. **الثقافة والسياحة:** نجد على شبكة WWW متاحف تعـــرض قســم مــن المجموعات الأثرية و المعارض المؤقتة و معلومات عامة مختلفة(مثل التوقيـــت، مخطط السير ضمن المعرض التعرفة... الخ

5. **التعليم والبحث:** يستعمل المدرسون و الباحثون الانترنت ليتبـــادلوا بســـرعة كبيرة أعمالهم، مثل نتائج تجاربهم او مصادرهم المعلوماتية والتربوية. كمـــا ان الانترنت هي مصدر هائل للمعلومات للمدرسين والطلاب، حيث يســـتطيعون الحصول على المعلومات المناسبة وهم وراء اجهزتهم.

6. **الألعاب:** يمكن لرائد الانترنت تحميل العاب عن بعد عبر البرامج: Freeware ., Shareware

(6-7-4) الاتصالات والعنونة على شبكة الانترنت:

بروتوكولات الاتصالات Communication Protocols:

ان الحديث بين شخص وآخر هو نوع من الاتصال، ولكي يتم الاتصـــل بطريقة صحيحة لا بد من مراعاة عدة قواعد، مثل ان تكون لغة التخاطب معروفـــة لكلا الطرفين، عندما يتكلم احد الطرفين لابد ان ينصت الآخر.. الخ هذا النـــوع من القواعد ينطبق أيضا على الاتصال بين الحاسبات، فعنـــد اتصـــال جـــهازي حاسوب معا، لابد أن تحكم اتصالهما مجموعة من القواعد لكـــي يتـــم الاتصـــال

بنجاح وهذه المجموعة من القواعد تسمى بروتوكولات الاتصال Communication protocols.

إن الاختلاف بين الحاسبات المتصلة بشبكة الانترنت وبين انظمة تشغيلها وبرامجها، وبين الكابلات الموصولة عليها، أدى الى ايجاد بروتوكول للتحكم في الاتصالات وهذا ما أطلق عليه البروتوكول القياسي لاتصال الانترنت (Transmission Control Protocol/Internet Protocol)(TCP/IP)

كما يوجد هناك العديد من البروتوكولات نذكر منها:

HTTP: بروتوكول نقل النصوص(الصفحات الإعلامية) Hypertext Transfer Protocol

FTP : بروتوكول نقل الملفات File Transfer Protocol

SMTP: بروتوكول البريد الالكتروني Simple Mail Transfer Protocol

العنونة على شبكة الانترنت Internet Addressing

من المعروف انه للاتصال بأي شخص آخر تلفونيا لابد من معرفة رقمه، وكذلك الحال بالنسبة للحاسبات، اذ لا يمكن الاتصال بأي حاسوب دون معرفة عنوانه. لكل حاسب متصل مع شبكة الانترنت عنوان هذا العنوان يسمى Ip Addresses وهو عبارة عن رقم مقسم الى أربعة أقسام A,B,C and B وكل قسم يأخذ أحد الأرقام بين الرقم صفر والرقم 255 ولكن الرقم صفر والرقم 255 محجوزان دائما لأغراض خاصة بالشبكة مثال هذه الارقام:

- 160.121.6.4
- 162.122.8.5
- 180.202.2.3

ونظرا لصعوبة حفظ هذه الأرقام تم استخدام الأسماء أو الأحرف بدلا منها وهذه العناوين في حقيقتها ليست اسماء الا لدى المستخدم، اما بالنسبة للحاسوب فهي ما زالت أرقاما على سبيل المثال.

العنوان بالنسبة للمستخدم Ritsec.com.eg

يعني بالنسبة للحاسوب 163.121.2.3

كما أن لهذه الاسماء أيضا قواعد لابد من إتباعها وهي كالتالي:

1. لا بد أن يكون اسم الدولة الموجودة بها الحاسب وهو جزء من عنوان الحاسب، لذلك تم اختصار اسماء الدول الى حرفين مثل Egypt تاخذ Eg و United Arab Emirates تاخذ ae وFrance تأخذ Fr.
2. تحت اسم الدول يأتي اسم نطاق العمل وكل نطاق يتم اختصاره الى حرفين او ثلاثة أحرف على سبيل المثال:
   - نطاق تجاري Commercial Sector اختصاره COM.
   - نطاق حكومي Government Organization اختصاره Gov.
   - نطاق تعليمي Educational Institutions اختصاره Edu.
   - الاغراض العسكرية Military اختصاره Mil.
3. تحت اسم نطاق العمل يأتي أسم الشركة أو المكان التابع لهذا النطاق على سبيل المثال IBM اختصاره ibm.
4. تحت اسم الشركة ياتي اسم الحاسوب مثال ذلك:

Mustafa.cf.gov.ae

Daher@Cultwral.org.eg

لابد من الاشارة هنا الى ان الرمز @ ينطق at(آت) و النقطة(.) تنطق Dot (دوت) و باقي الحروف كما هي.

(6-7-5) أسس الدخول الى شبكة الانترنت

من اجل الدخول على شبكة الانترنت لا بد من توفر العناصر الضرورية التالية:

1. حاسوب شخصي طول الكلمة فيه 32 بت، مزود بجهاز مودم Modem ومزود أيضا بخط خاص مباشر(Leased Line).
2. أسم تعريفي وكلمة سر (اشتراك في الانترنت).

3. برامج تتيح الدخول على خدمة الانترنت.

يمكن تقسيم طرق الدخول على الانترنت الى طريقتين:

1. طريقة غير مباشرة:

وهي الطريقة التي يسلكها معظم مشتركي الانترنت إذ يكون الحاسوب الشخصي مزودا بفرع Terminal يربطه بالحاسوب الرئيسي يكون متصلا مباشرة بالانترنت.

2. طريقة مباشرة:

وهي أن يكون الحاسوب الشخصي عقدة مستقلة Individual Node على شبكة الانترنت، ويملك القدرة على ادخال أي مشتركين آخرين على الشبكة عن طريقة.

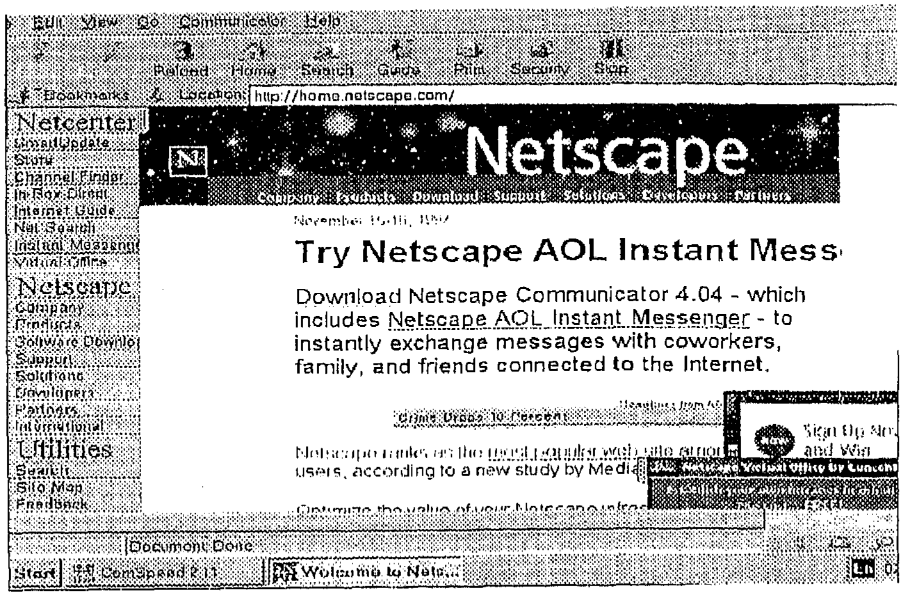
(6–7–6) البرامج المستخدمة في البحث وتصفح المعلومات عبر الانترنت Internet Browsers

من أهم البرامج المستخدمة لعرض و البحث عن المعلومات هي:

1. برنامج نتسكيب Netscape
2. برنامج موزايك Mosaic
3. برنامج اكسبلورير Explorer

برنامج نتسكيب Netscape:

هذا البرنامج يحتوي على تقنيات يمكن من خلالها الدخول الى شبكة الانترنت ويمكن من خلاله ارسال و استقبال الرسائل والأخبار والمعلومات، الشكل(5) يوضح الشاشة الرئيسية لهذا البرنامج.



الشكل(5) الشاشة الرئيسية لبرنامج **Netscape**

مجموعة مفاتيح برنامج Netscape:

- المفتاح Back: يرجع الصفحة الاعلامية السابقة.
- المفتاح Forward:يتقدم الى صفحة تالية.
- المفتاح Home:يظهر صفحة بداية تشغيل برنامج نتسكيب
- المفتاح Reload: خاص باعادة تحميل الصفحة مرة اخرى.
- المفتاح Image: خاص باعادة تحميل الصور داخـــل الصفحـــة اذا لم تكـــن واضحة عند تحميلها في المرة الاولى.
- المفتاح Open: خاص بفتح صفحة اعلامية اخرى.
- المفتاح print: يقوم بارسال الصفحة التي امامك الى الطابعة.
- المفتاح Find:خاص للبحث عن كلمة داخل الصفحة الموجودة فيها.
- المفتاح Stop:خاص لعدم اكمال تحميل الصفحة.
- المفتاح What's New : من خلال هذا الزر يستطيع المستخدم معرفة ما هـــي الاماكن الجديدة التي تم افتتاحها على شبكة الانترنت.

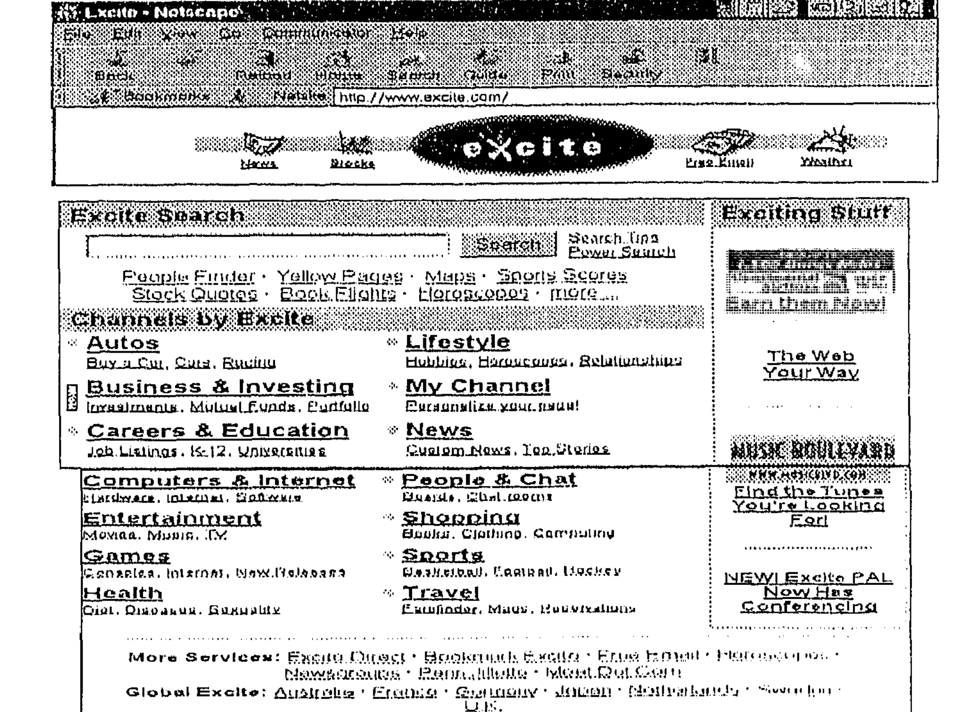
- المفتاح Whats Cool: لتثبيت الاماكن التي يكثر استخدامها على الشاشة.
- المفتاح Destinations: يساعد في الحصول على اكثر الاماكن التجارية التي يستخدمها رجال الاعمال على شبكة الانترنت.
- المفتاح Net search: يعرض اكثر من طريقة للبحث عن موضوع معين.
- المفتاح People: خاص بالصفحات البيضاء White Pages، حيث بواسطته نستطيع البحث عن أي مستخدم داخل الشبكة بواسطة اسمه او عنوانه البريدي.
- المفتاح Soft Ware: خاص ببرنامج Netscape وذلك من اجل الحصول على الاصدار الجديد للبرنامج او الاطلاع على منتجات شركة Netscape.
- مستطيل العناوين Net site: يوضح هذا المستطيل عنوان الصفحة التي تظهر امامك على الشاشة، اذا اردت ان تنتقل الى صفحة اخرى او الى أي مكان اخر، تستطيع كتابة العنوان داخل هذا المستطيل ثم Enter.

يمكن استخدام عدة انظمة من اجل البحث عن المعلومات في الانترنت باستخدام برنامج Netscape من اهمها:

أ. اكسايت **Excite**:

توفر هذا النظام شركة Architext Software، ويشمل أداتين للبحث، الاداة الاولى Net Search وهي تتيح المجال للبحث عن مواقع معينة في الشبكة او البحث في المجموعات الاخبارية Newsgroups لشبكة Usenet الأداة الثانية Net Reviews وهي توفر مخطط هرمي لمحتويات الشبكة مفهرسة حسب الموضوعات و تفرعاتها يمكن الوصول الى موقع Excite من خلال العنوان التالي:
http://www.excite.com

والشكل (6) يبين الشاشة الرئيسية لنظام Excite.



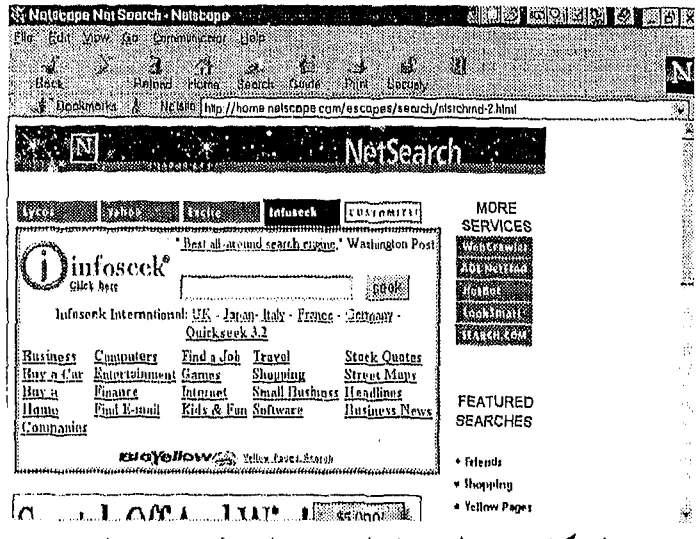
الشكل (6) الشاشة الرئيسية لنظام **Excite**.

ب. نظام انفوسيك **Infoseek**:

يوفر هذا النظام الخدمة على شكلين، شكل مجاني و توفر هذا الشكل من الخدمة شبكة Usenet Web، وشكل مدفوع الآجر، يمكن الدخول علـــــى نظـــام انفوسيك باستخدام العنوان التالي:

http://www.infoseek.com

والشكل(7) يبين الشاشة الرئيسية لنظام انفوسيك ضمن برنامج Netscape



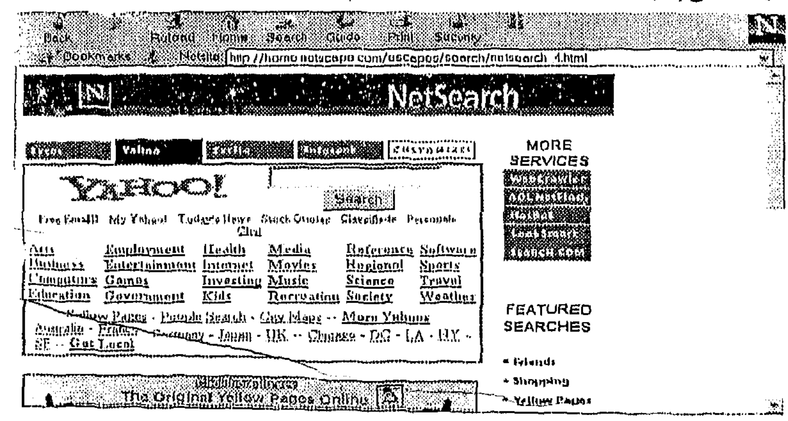
الشكل (7) الشاشة الرئيسية في نظام انفوسيك

جـ نظام خدمة ياهو Yahoo

وهو اكثر الأنظمة استخداما، حيث يحتوي على آليات بحث متطورة لأنه يحتوي على كلمات مفتاحية وفهرس منظم حسب المواضيع، ويحتوي أيضا على دليل يصف المواقع الأكثر شيوعا عند المستخدمين، يمكن الوصول الى نظام ياهو من خلال العنوان التالي:

http://www.Yahoo.com

والشكل(8) يبين الشاشة الرئيسية لنظام ياهو



الشكل (8) الشاشة الرئيسية في نظام ياهو

د. نظام خدمة ليكوس **Lycos**

يعد هذا النظام من أهم آليات البحث المتوفرة على الشبكة، حيث يتوفر فيه قاموس يشمل اهم الكلمات والمصطلحات في الشبكة تصل الى 8 ملايين كلمة، ويضم أيضا آليات للبحث عن النصوص والصور و الصوت، يمكن الوصول الى موقع ليكوس من خلال العنوان التالي:

http://ww.lycos.com

هـ نظام خدمة ماجلان **Magellan**

يحتوي هذا النظام قاموسا يحتوي على أهم العناوين على شبكات Gopher, FTB , Web ,Usenet. وعنوان هذا النظام هو:

http://www.mckinley.com

و. نظام خدمة فهرس السياحة **Web Servers Tourist Directory**

يشمل هذا النظام تقنية متطورة لمتابعة المواقع العلمية والسياحية على خارطة العالم المعروضة على الشاشة، وعنوان هذا النظام:

http://www.w3.org/hypertext/datasawrces/www/servers.html

أسئلة وتمارين عامة

1. عرف الشبكة و ما هي أنواع الشبكات؟
2. ما هي أهم معدات الشبكات؟
3. ما هي أهم وحدات عمليات ارسال و استقبال البيانات؟
4. ما هي أهم قنوات الاتصال و ما هي أهم أنواع القنوات حسب تراكيبها الفيزيائي؟
5. اذكر أهم الشبكات المحلية للحاسوب؟
6. عرف شبكة الانترنت؟
7. اذكر أهم خدمات شبكة الانترنت؟
8. عرف بروتوكولات الاتصال واذكر اهمها؟
9. كيف تتم عملية العنونة على شبكة الانترنت؟
10. ماذا يلزمنا من اجل الدخول على شبكة الانترنت؟
11. تحدث عن الطريقة المباشرة و الغير مباشرة للدخول على شبكة الانترنت
12. ما هي أهم البرامج المستخدمة في البحث و تصفح المعلومات في الانترنت؟
13. ما هي أهم الأنظمة المستخدمة في برنامج نتسكيب؟

الفصل السابع

لغات البرمجة

Programming Languages

(7 – 1) المكونات والعناصر الأساسية للغة البرمجة :

اللغة البرمجية هي مجموعة من القواعد والاصطلاحات تستخدم لتمثيل المعلومات من قبل الانسان ويفهمها الحاسوب بعد ترجمتها بواسطة مترجمات او مفسرات الى لغة الآلة ، وتتكون لغة البرمجة مثل أي لغة انسانية من مجموعة من العناصر التالية : Beginner's All - Purppose Symblic Instruction code

(7–1–1) مجموعة الرموز الاساسية :

وهي مجموعة العناصر التي تكون اللغة ، وهي مكونة من كل أو جزء من الرموز آلاتية :-

1- الحروف اللاتينية وهي :

A ,B , C, D , E,... X, Y , Z

2- الأرقام العربية و هي :

0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9

3- الرموز الخاصة وهي :

+ , - , * , ; . $, (,) , { , } , < , > , = , ?, ...

(7-1-2) الكلمات

وهي نوعان :

أ- **الكلمات المحجوزة :** وهي كلمات لها معاني خاصة بالنسبة للحاسوب وعندما يتعرف الحاسوب عليها يقوم بتنفيذ الاوامر المناظرة لمعاني هذه الكلمات ومن امثلتها : ... , Read , Print , FOr

لا يجوز للمبرمج اعادة تعريفها لأي غرض آخر في البرنامج .

ب- **مسميات المبرمج التعريفية :** وهي كلمات يختارها المبرمج في البرنامج لعنونة البيانات مثل :sum, I,Z, qty,...

وتختلف قواعد كتابة مسميات المبرمج من لغة الى أخرى .

(7-1-3) مجموعة التعليمات والأوامر :

التعليمة : هي مجموعة من الرموز الخاصة بلغة البرمجة التي عندما يفهم الحاسوب معناها ومدلولاتها تستطيع وحدة التحكم فيه أن ترسل الاشارات التنظيمية لبقية وحدات الحاسوب القيام بما تمليه عليها مفهوم هذه التعليمات ، وتنقسم مجموعة التعليمات أو الأوامر التي تتضمنها أية لغة برمجية من الناحية الوظيفية الى الانواع التالية :

1. تعليمات وأوامر الإدخال : وهي التي تقوم بقراءة أو إدخال البيانات المدخلة وتخزينها في وحدة التخزين الاساسية عن طريق وحدة الإدخال مثل INPUT, READ, PRINT.
2. تعليمات وأوامر تداول البيانات : وهي التعليمات التي تقوم بعمليات معالجة البيانات داخل الحاسوب مستخدمة وحدة الحساب والمنطق .
3. تعليمات واوامر التحكم : وتشمل التعليمات التي تؤدي الى تغير تسلسل تنفيذ جمل البرنامج نتيجة إجراء اختبار معين باستخدام وحدة الحساب والمنطق .

4. تعليمات وأوامر الإخراج :وهي التعليمات التي تقـوم بـاخراج البيانـات المعالجة على شكل معلومات من الذاكرة الرئيسـية الى وسـط الاخـراج بواسطة وحدات الإخراج مثل : الطابعة الشاشة

5. تعليمات تعريفية : وهي التعليمات التي تقوم بتعريف الحاسوب على :

أ. مواضع التخزين الرئيسية .

ب. نوع البيانات التي يمكن أن تخزن في المواضع التخزينية وهـي ثلاثـة أنواع: رقمية ، حرفية ، حرفية ورقمية .

جـ أنواع الوحدات المستخدمة في البرامج مثل نوع وحـدات الادخـال والاخراج .

(7-1-4) قواعد اللغة :

وهي مجموعة القواعد والقيود التي يجب ان نلتزم بها عند كتابة البرنـامج حتى يستطيع الحاسوب تنفيذ تعليمات وأوامر البرامج المكتوبة بهذه اللغة .

(7 2) أنواع لغات البرمجة

تصنف لغات برمجة الحاسوب الى مستويين اساسيين : لغات المستوى الواطئ ولغات المستوى العالي .

(7-2-1) لغات المستوى الواطئ

وتنقسم أيضا الى نوعين هما لغة الآلة و لغة التجميع :

1. **لغة الآلة Machine Language**: وهي اللغة الوحيـدة الـتي يفهمـها الحاسوب مباشرة دون وسيط ، وتعليمات هذه اللغة هي مجموعة من الأرقـام الثنائية إما (1 , 0) ظهرت هذه اللغة مع بداية ظهور الحاسوب وكـان المبرمجون بلغة الآلة يحتاجون الى معرفة مكونات الحاسوب وامكانياته ، مما أدى الى صعوبة فهم تلك اللغة وخاصة أن لكل حاسوب لغة آلـة خاصـة بـه ،

وبالتالي لا يمكن نقل البرنامج من حاسوب معين الى حاسوب آخر لذلك اقتصر استخدامها على الشركات الصانعة للحواسيب .

2. **لغة التجميع او اللغة الرمزية : Assembly Language** : وهي مرحلة متقدمة من لغة الآلة واسهل نسبياً منها ، وقد قامت الشركات المنتجة للحواسيب بتصميمها للتغلب على صعوبة استخدام لغة الآلة حتى تساعد على انتشارة الحواسيب وقد تم استبدال الأرقام الثنائية برمز وهو عبارة عن حرفين أو ثلاثة أحرف يكون اسهل في تذكرها وكتابتها ويطلق عليها (Mnemonic) ، وتختلف الرموز المستخدمة باختلاف طراز لغة التجميع والشركة المنتجة له. وتعتبر هذه اللغة مرحلة وسط بين لغة الآلة واللغات ذات المستوى العالي ، وتحتاج لغة التجميع الى برنامج مترجم لترجمتها الى لغة الآلة التي يفهمها الحاسوب . ومن ابرز عيوب لغة التجميع هو ارتباطها بالآلة ، فكل آلة لها لغة التجميع الخاصة بها ، ويقصد بالآلة هنا تحديدا المعالج Processor او المعالج الدقيق Microprocessor .

7 2 2) **لغات المستوى العالي** : High - level Languages

وقد سميت بهذا الاسم لبعدها عن لغة الآلة ، ولايتطلب من المبرمجين المعرفة بالتفاصيل الدقيقة لجهاز الحاسوب كما هو الحال بالنسبة للغات ذات المستوى المنخفض و لكن يتطلب منه الالمام العام باسلوب عمل الحاسوب ومنطقه ، أما المعرفة التفصيلية فتكون لجوانب المشكلات والمسائل الخاصة ، بالتطبيقات ويمكن ايجاز مميزات اللغات ذات المستوى العالي فيما يلي :

1. عدم ارتباطها بآلة معينة مثل لغة الآلة ولغة التجميع .
2. سهولة نقلها وسهولة كتابة البرامج بها وذلك لاستخدامها كلمات وتعابير مشابهة لتلك التي يستخدمها الانسان في حياته العادية .
3. سهولة اكتشاف الأخطاء وتصحيحها .

4. لا تحتاج عملية تغيير الحاسوب بحاسوب آخر الى تغيير كبير في البرامج وذلـــك لان اللغة مصممة أساساً لحل نوعية معينة من المشاكل وليست لنوعية معينـــــة من المعالجات .

5. توفير الوقت والجهد للمبرمجين مقارنة مع الوقت والجهد المبـــذول باســـتخدام لغات المستوى المنخفض .

6. سهولة التعامل مع الحاسوب .

وتنقسم لغات هذا المستوى الى عدة مستويات اخرى يمكن تصنيفها كما يلي :

1 – لغات مرتبطة بالاسلوب	Procedure-Oriented Languages
2 – لغات مرتبطة بالمشكلة	Problem-Oriented Languages
3 – لغات استفسارية	Query Languges Languges
4 – لغات الذكاء الصناعي	Artificial Intelligence Languages
5 – لغات البرمجة الشيئية	Object-Oriented Languages

وفيما يلي نبذة عن اهم لغات كل مستوى من المستويات السابقة .

(7–2–2–1) لغات مرتبطة بالاسلوب

Procedure-Oriented Languages

هذه اللغات تتسم بالمرونة ويوجد منها لغات متعددة وتنقسم من حيـــث الاستخدام الى :

1. لغات للتطبيقات التجارية وإدارة الأعمال
2. لغات للتطبيقات العلمية
3. لغات عامة الاغراض

أولا : لغات التطبيقات التجارية وادارة الاعمال :

1) لغة الكوبول (Cobol)

اشتقت كلمة كوبول من الحـــروف الاولى للعبـــارة :

Common Business Oriented Language

وتعني اللغة الموجهة للاعمال التجارية العامة ، بدا ظهورها نهاية عام 1959 بدعم من وزارة الدفاع الامريكية . ثم تطورت في صور كثيرة حتى استقرت على آخر صورة لها عام 1985 (Cobol 85).

من مزايا لغة (Cobol) هي :

1. استقلالها عن جهاز الحاسوب اذ يمكن استخدامها على أنـــواع مختلفـــة مـــن الحواسيب .
2. اعتمادها على أصول قياسية موحدة .
3. قدرتها على التحكم في البيانات ، و تنويع اساليب معالجتها .
4. قدرتها على تنظيم الملفات و توثيقها .
5. انتشارها الواسع في مجال التطبيقات التجارية .
6. يمكن استعمال اسماء و جمل و عبارات قد تصل طولها الى (30)-حرفاً ، الأمـــر الذي تفتقد له لغة البيسك والفورتران .

اما عيوب لغة (Cobol) فهي :

1. طول برامجها مقارنة مع لغات اخرى ، وهذا قد يسبب الملل للمبرمج و فقـــدان السيطرة على اجزاء البرنامج .
2. قصورها عن برمجة المعادلات الرياضية و المشاكل الاقتصادية المعقدة .
3. بسبب تعاملها مع مجموعات كبيرة من الملفات والبيانات فانها تستهلك مساحة كبيرة من الذاكرة في الحاسوب .

2) لغة أر بي جي (RPG)

وهذه اللغة اشتقت كلمة أر بي جي من الحروف الاولى للعبارة التالية :

Report Program Generator وتعني مولد برامج التقارير ، استحدثت هـذه اللغة عام 1946 من قبل شركة IBM ثم طورتها عام 1970 لحل مسائل مختلفـــة وخاصة التجارية منها ، وما يتعلق بتحديث الملفات ، إذ أن الأعمـــال التجاريـــة

تحتاج دائما الى إعداد الوثائق والشيكات والتقارير العادية وهذه اللغـــة صممـــت أساساً لتحقيق هذا الهدف بالاضافة الى انها لغة سهلة التعلم .

من عيوب لغة (RPG) :

1. لغة محدودة الانتشار فهي خاصة ببعض الحواسيب .
2. امكانياتها الرياضية محدودة جداً بالرغم من أنه هناك إصدارت جديدة مـــن PRG تستطيع التعامل مع المصفوفات .

ثانياً : لغات التطبيقات العلمية :

1) لغة فورتران Fortran

وهي اختصار لـــ(FORmula TRANslator) أي مترجم معادلات ، وهي لغة علمية هندسية من اقدم لغات المستوى العالي ، وقد وضعـــت أصولهـــا شركة IBM عام 1957.

من ميزات لغة (Fortran) أنها :

1. متخصصة في برمجة المسائل العلمية والهندسية .
2. ذات تعليمات واصطلاحات مختصرة وسهلة.
3. لغـــة متطـــورة ، واصدارتهـــا الاخـــيرة فورتـــران 77 (77 Fortran) و Fortran 90 يضم اضافات وخصـــائص لم تكـــن موجـــودة سابقـــا ، حتى أنه اصبح بالامكان استخدامها لبرمجة المســـائل التجاريـــة ، وعمليـــات البرمجة الرمزية والحرفيـــة .

اما عيوب لغة (Fortran) فهي :

1. تركيبـــــها لا يســـــــمح باســـــــتخدام البرمجـــــــة الهيكليـــــــة Structured Programming .
2. اعتمادها على ترميز الاسطر ضمن اعمدة محددة مما يجعلها لا تناسب اولئـــك الذين اعتادوا على البرامج الحرة في الشكل .

2) لغة ألجول Algol

اشـــتق اسمــــها مــــن كلمـــتي ALGOrtimic Language أي لغـــــة الخوارزميات، ظهرت عام 1958 وكانت تهدف الى حل المسائل العلمية والعددية، سميت في ذلك الوقت Alogo 58 وتم تنقيحها عام 1960 ، وتتميز بالمنهجيـــة ، والهيكلية والدقة في تعريف اللغة و تحديد اساليب تخطيط البرامج ، وهذه اللغة اكثر شيوعاً في اوروبا على عكس لغة الفورتران المنتشرة في امريكا اكثر مـــن اوروبـــا بسبب سيطرة شركة IBM في امريكا و التي تفضل لغة الفورتران .

3) لغة أبل APL

صممت عام 1962 من اجل العمل بنظـــام المشاركة الزمنية ، و لم يســـمح لها بالاستخدام العام الا في عام 1968 ، و تعتبر لغـــة ســـهلة التعلـــم وتشـــتهر باستعمالها لأغراض المعالجة الجدولية Table Handling .

من عيوب لغة APL :

1. انها تستخدم حروفا لاتينية لبعض تعليماتها .
2. عدد تعليماتها كبير مقارنة مع لغات اخرى.
3. تحتاج الى مترجم كبير وذاكرة واسعة جدا في الحاسوب .

ثالثا : لغات عامة الاغراض :

1) لغة بيسك Basic

كلمـــــــة Basic اختصـــــــارا للكلمـــــــات التاليـــــــــــة:

وتعـــني

كود التعليمات الرمزية المتعددة الاغراض للمبتدئين ، وضعت أصول هذه اللغـة في كلية دارتموث الأمريكية على يدي كل من جون كيمي وتوماس كيرتز عام 1965 وكان الهدف من وضعها ان تكون لغة تعليمية للمدارس والجامعات ، وفي اواخـــر

الستينات تطور استخدامها بسبب ادخال نظام المشاركة الزمنية الى بيئة الحاسوب فتعدت حدود المدارس والجامعات لتشمل الاعمال التجارية .

من أهم ميزات لغة (Basic) :

1. سهولة تعليماتها وسرعة فهم رموزها واصطلاحاتها
2. تستخدم لاغراض متعددة منها التعليمية ومنها التجارية .
3. تطوير الشركات العالمية المنتجة للبرامج لها باستمرار ، وقد ظهرت عدة اصدارات للغة البيسك :

Basic - Gwbasic - Turbo Basic - Quick Basic - Visual Basic

4. توفرها في معظم الحواسيب الشخصية .
5. سرعة استجابة وتنفيذ البرامج عند ادخالها ، اذ ان هذه اللغة تعمل على نظام الترجمة المباشرة Interpretation وهو نظام يوفر فرصة التخاطب مع الحاسوب بحيث يتم اكتشاف اخطاء البرامج بسرعة و يتم اصلاحها .

2) لغة باسكال PASCAL

سميت هذه اللغة باسم عالم الرياضيات الفرنسي باسكال ، وقد صممت هذه اللغة في أواخر الستينات واوائل السبعينات من قبل البروفيسور نيكولاس ويرث في المعهد الفيدرالي السويسري للتكنلوجيا ، وهي أول لغة رئيسية تم تصميمها بعد وضوح السمات الاساسية للبرمجة الهيكلية .

من مزايا لغة (Pascal) :

1. أنها تتفوق على اللغات الراقية الاخرى باعتمادها مفهوم البرمجة الهيكلية ، ذات الترتيب المنطقي المتسلسل .
2. يمكن استعمالها في المجالات الرياضية ، والعلمية والهندسية والتجارية والادارية كما انها لغة تعليم جامعي .
3. تجمع بين مزايا الفورتران ومزايا البيسك ومزايا لغة الجوال .
4. برامجها قصيرة اذا ما قيست ببرامج الكوبول في التطبيقات التجارية .

3) لغة بي ال واحد (PL/1)

مشتقة من العبارة (Program Language / One) صممت هـذه اللغة في منتصف الستينات في محاولة لتجميع خصائص معظم اللغـــات وخاصـة الفورتران و الكوبول في لغة واحدة لدى شركة IBM الامريكية ،وقد ظهرت لغة Pl/1 القياسية طبقا للقياســات العالميـة ANSE في عــام 1976، وبعــض الاصدارات من هذه اللغة القياسية اصبح متاحا على الحواسيب الشخصية .

من ميزات لغة (PL/1) :

1. سهولة تعلمها ، وسرعة كتابة البرامج بها لأنها لغة اشد اختصار من غيرها .
2. تجمع بين الاغراض الهندسية و العملية و بين الاغراض التجارية .
3. تستعمل اسلوب البرمجة الهيكلية .

من عيوب لغة (PL/1) :

1. تكاد أن تكون لغة خاصة بشركة IBM ، فهي محدودة الانتشار .
2. تشبه الكوبول في حاجتها الى مساحات تخزينية واسعة في ذاكرة الحاسوب .

4) لغة أدا ADA :

سميت هذه اللغة باسم اول مبرمجة وهي اجستا أدا بايرون ، صممت لغـة أدا في وزارة الدفاع الامريكية عام 1980 بهدف ان تكون لغة قياســية للأنظمـة العسكرية ثم استخدمت بعد ذلك للاغراض التجارية ، وتعتبر لغة (أدا) لغة البرمجـة الهيكلية تساعد في تصميم المنظومات الهندسية جزءا جزءا ، ولديها القـدرة علـى فحص البرامج والتأكد من خلوها من الاخطاء قبل تنفيذها ، إلا انها تحتاج الى سعة ذاكرة كبيرة ، مما يجعل استخدامها على الحواسيب الصغيرة أمرا صعبا .

5) لغة سي C :

انتجت هذه اللغة في معامل بل في اوائل السبعينات واستخدمت لكتابة الانتاج الاول لنظام التشغيل يونيكس ، وتوجد اصدارات متعددة للغة الـ سي تعمل مع الحواسيب الشخصية والحواسيب الكبيرة والمتوسطة .

من ميزات لغة C :

1. تعتبر هذه اللغة وسطا بين اللغات الراقية كالبيسك والفورتران والكوبول والباسكال وبين لغة التجميع ، فهي تحل محل لغة التجميع في عملياتها ، وفي الوقت نفسه تؤدي اعمال اللغات الراقية ، لذلك انتشر استخدامها في الاوانة الاخيرة بشكل ملحوظ .
2. تعتبر من لغات البرمجة الهيكلية مثل الباسكال .
3. يمكن تحويل البرامج المكتوبة بلغة سي من نوع من الحواسيب الى نوع اخر دون مشقة .

(7-2 2 2) لغات مرتبطة بالمشكلة Problem - Oriented Langues

وهي لغات صممت لحل مشاكل معينة او تطبيقات خاصة ، وهي لا تتطلب عادة برمجة تفصيلية كالغات المرتبطة بالاسلوب وانما يكون التركيز اكثر على المدخلات والمخرجات ومن امثلة تلك اللغات :

1) لغة لوجو LOGO :

تم إنتاجها في أواخر الستينات على يد سيمون بايت وزملائه في معهد (MET) بامريكا وقد صممت خصيصا للاطفال فهي لغة بسيطة وسهلة التعلم ويسهل بواسطتها عمل الرسوم وتلوينها وتحريكها وتستخدم أيضا في المدارس لتأليف المقاطع الموسيقية وادارة البيانات والتحكم في النصوص ،بالاضافة الى انها تعمل على الحواسيب الشخصية .

2) لغة بيلوت PILOT :

وهي اختصار للعبارة Programmed Inquiry Learning or Teaching أي الاستفسار و التعليم او التدريس المبرمج ، وقد صممت لكتابة البرامج التعليمية فيمكن بواسطتها كتابة الاسئلة ثم اجابتها ورد فعل الحاسوب في حالة الصواب أو الخطأ كما يمكن بسهولة عمل الرسومات المختلفة بها .

(7-2-2-3) لغات استفسارية Query Languages

صممت هذه اللغات أساسا للمدراء . وهي عبارة عن تعليمات باللغة الانكليزية تماما و لا تحتوي الا على عمليات حسابية محدودة وتتميز بقدرتها على عمل التقارير وترتيب واختيار البطاقات والسجلات بطريقة آلية وهو ما يهم المدراء.

(7-2-2-4) لغات الذكاء الاصطناعي

Artificial Intelligence Languages

1) لغة ليسب LISP

وهي اختصار (List Programming Language) أي لغة برمجة القوائم ، صممها جون مكارثي (1959-1960) لتدعيم البحوث في مجال الذكاء الاصطناعي حيث للتحكم في البيانات غير الرقمية وقد ظلت حتى الان اللغة المفضلة في الولايات المتحدة الامريكية للباحثين في مجال الذكاء الاصطناعي.

2) لغة برولوج PROLOG

كلمة Prolog اختصارا للعبارة (Programming in Logic) أي البرمجة المنطقية ، وقد تم تصميمها في فرنسا ، ثم اختارها اليابانيين لتكون اللغة القياسية للجيل الخامس للحواسيب (وهو المشروع الذي خططت له ونفذته اليابان لأنتاج حواسيب ذكية تحاكي الانسان في قدراته على التحليل والاستنتاج)، بالاضافة الى فعاليتها في مجال النظم الخبيرة القادرة على تخزين المعرفة و الخبرات المختلفة .

7-2-2-5) لغات البرمجة الشيئية

Object-Oriented Languages

وهي لغات تقوم على تعريف الأشياء المستخدمة بها ، و يكون تعريـــــف الشيء مكونا من حالة الشيء والمنهج (Method) أي يوصف العمليات الـــتي يمكن تنفيذها على هذا الشيء ، ومن امثلة لغات البرمجة الشيئية :

1. Smalltalk
2. C++
3. Visual C++
4. Pascal 5.5

أسئلة وتمارين عامة

1. عرف اللغة البرمجية ؟
2. عدد مكونات اية لغة برمجية ؟
3. تكلم باختصار عن الفرق بين لغات المستوى العالي ولغات المستوى المنخفض؟
4. عدد انواع لغات المستوى العالي مع ذكر اسم احدى لغات كل نوع ؟
5. اذكر أهم ميزات لغة باسكال ؟

الفصل الثامن

برمجة الحاسوب والمخططات الإنسيابية

Computer Programming

تعتبر برمجة الحاسوب علماً وفناً في نفس الوقـــــت ، ولحـــل أي مشـــكلة باستخدام الحاسوب نتبع عددا من الخطوات ، وحسب درجـــــة كفائـــة هـــذه الخطوات تتحدد كفاءة تشغيل المسالة حاسوبياً ، و الجدير بالذكر ان اهـــم هـــذه الخطوات يتم تنفيذها خارج الحاسوب وبدون استخدام تقنيات الحاســـوب ، إذ إنها تمثل منطق حل المسألة.

لذلك يمكن تقسيم عملية اعداد برنامج يتم تطبيقه على الحاسوب إلى ثـلاث مراحل اساسية و كل مرحلة تتضمن مجموعة من الخطوات ،تؤدي جميعها بشـــكل متكامل الى الوصول الى برنامج حاسوبي للمشكلة المطلوب معالجتـــها باســـتخدام الحاسوب .

(8 –1) المراحل الأساسية لحل أي مسألة حاسوبيا :

تتكون المراحل الأساسية لحل أية مسألة باستخدام الحاسوب من ثلاثة مراحل وهـــي كما يلي :

(8 – 1 – 1) المرحلة الاولى (دور الإنسان) :

إن دور الإنسان في هذه المرحلة تعتمد على المختص بالمشـــكلة وكذلـــك تعتمد على محلل النظم ومن ثم المبرمج وتتضمن هذه المرحلة الخطوات التالية :

الخطوة الأولى : تعريف المسالة وتحليلها

Problem Definition and Analysis

تعتبر هذه الخطوة من أهم وأصعب الخطوات على الاطلاق حيث يتم فيها تحديد أبعاد المسألة ، وتحديد الهدف المطلوب الوصول اليه وذلك بتحليل مفردات المسالة ووصفها ومن بجموعة من المهام وهي :

1. تحديد أهداف المسألة و تحديد النتائج المراد تحقيقها من حيث نوع المعلومات واسلوب عرضها وشكل المعلومات ، والتي تدعى المخرجات لنتمكن من تحديد المدخلات والعمليات اللازمة لتحقيقها ، فالنتائج تحدد وتعرف أولاً ونحصل عليها بعد إجراء المعالجة وإظهار النتائج .
2. تحديد المدخلات اللازمة للحصول على المخرجات المعرفة مسبقاً، وكذلك تحديد شكل المدخلات ومواصفاتها بدقة .
3. حصر طرق الحل المختلفة من وجهة نظر الحاسوب ، وتقييمها لاختيار الحل الأفضل والأكثر ملائمة للتنفيذ باستخدام الحاسوب من حيث سهولتها وسرعة تنفيذها والمساحة التي تحتاجها في ذاكرة الحاسوب .

ويشترك في اعداد متطلبات هذه الخطوة الإحصائي في موضوع المشكلة المدروسة (المحاسب، المهندس ، ...) بالاضافة الى محلل النظم المختص بعلم الحاسوب لتهيئة المسألة ووضع أسلوب حل يتناسب مع منطق الحاسوب .

الخطوة الثانية : وضع خوارزمية الحل Algorithm

بعد اختيار الطريقة المثلى لحل المسألة قيد الدراسة في الخطوة السابقة نقوم بالتعبير عن أسلوب حل المسألة على شكل خطوات متسلسلة و متعاقبة و مترابطة منطقيا تؤدي الى الوصول الى حل المسالة. يدعى هذا التسلسل من الخطوات بالخوارزمية Algorithm ، و التي يمكن تعريفها كما يلي :

الخوارزمية (*): هي عدد من الأوامر أو التعليمات المترابطة والمرتبة على شكل خطوات متعاقبة، يمكن تنفيذ كل منها آلياً من أجل الوصول الى حل المسألة، وبشكل عام يمكن ان نطلق كلمة الخوارزمية على طريقة الحل اذا توفرت في هذه الخوارزمية النقاط التالية :

1. أن تحوي هذه الخوارزمية على عدد محدد من الخطوات للتوصل الى النتيجة، بمعنى آخر لا يمكن تسمية طريقة حل ما بالخوارزمية إذا حوت عدد غير منتهي من الخطوات .
2. يجب أن تكون الطريقة واضحة ، وتتضمن تعليمات مفصلة و صريحة و أن تكون كل خطوة واضحة بشكل لا لبس فيه .
3. أن تحوي الخوارزمية على طريقة لجميع الطريقة جميع الحالات الخاصة والتي يمكن ان تعترض الانسان خلال عملية الحل .
4. أن تكون الخوارزمية صالحة لحل جميع المسائل المشابه والتي تنطبق عليها نفس الشروط .

مثال (1) : اكتب خوارزمية ايجاد مساحة ومحيط دائرة نصف قطرها معلوم (R)

الخوارزمية :

الخطوة (1) : ادخل قيمة نصف القطر R

الخطوة (2) : اجعل قيمة PIE = 3.14

الخطوة (3) : احسب مساحة الدائرة حسب العلاقة : A = B * R * R

الخطوة (4) : احسب محيط الدائرة حسب العلاقة : C = 2 * B.* R

(*) استخدمت الخوارزميات في الماضي بشكل واسع في اوروبا وامريكا ، وكانت تعني الوصف الدقيق لتنفيذ مهمة من المهمات أوحل مسألة من المسائل، وقد اشتق اسم هذه الكلمة من أسم العالم الشهير الخوارزمي (ابو جعفر محمد بن موسى الخوارزمي) وهو اول من استخدم العدد صفر ، و صاحب كتاب (الجبر و المقابلة) أما الآن فقد ارتبط هذا الاصطلاح ارتباطا وثيقا بعلم الحاسوب حتى أن احد اقطاب علم الحاسوب هو الاستاذ khuth يعرف علم الحاسوب على أنه (دراسة الخوارزميات) .

الخطوة (5) : اطبع قيم كل من A , C , R حيث تمثل R : نصف القطر ، A : المساحة ، C : المحيط .

مثال (2) : اكتب خوارزمية لتحديد فيما اذا كانت قيمة A مساوية لقيمة B ام ان احدهما أكبر من الآخر وذلك باستخدام منطق الحاسوب .

الخوارزمية :

الخطوة الأولى : ادخل قيمة A , B

الخطوة الثانية : قارن A مع B ، اذا كانت A = B عندئذ اطبع A = B واذهب الى الخطوة الخامسة .

الخطوة الثالثة : اذا كانت A<B عندئذ اطبع A<B واذهب الى الخطوة الخامسة .

الخطوة الرابعة : اطبع A>B .

الخطوة الخامسة : اما ادخال قيم جديدة لــ A ,B والعـــودة الى الخطــوة الاولى او التوقف.

مثال (3) : ارادت شركة تجارية القيام بحساب الربح الصاف والـــذي يمثـــلNP لتوزيعه على الشركاء في نهاية العام، وذلك بطرح المصاريف المختلفة E من الربــح الاجمالي GP ، فاذا كان عدد الشركاء و هم متساوون في حصصهم هـــــو N ، فاكتب الخوارزمية المثالية لحل هذه المسألة .

الخوارزمية :

الخطوة الأولى : أدخال قيمة GP وقيمة E وقيمة N

الخطوة الثانية : احسب الربح الصافي من المعادلة : NP = GP - E

الخطوة الثالثة : احسب حصة كل شريك في الشركة من المعادلة P = NP / N

الخطوة الرابعة : اطبع النتيجة P , NP

الخطوة الثالثة : وضع المخطط الانسيابي Flowchart

ان المخطط الانسيابي(*) (خريطة سير العمليات) هو التعبير عن الخوارزمية بالرسم وذلك باستخدام رموز واشكال اصطلاحية لمساعدة مخطط البرامج على وضع تصور مرئي لخطوات الحل اللازمة للمشكلة المدروسة بواسطة الحاسوب .

ولأهمية خرائط سير العمليات في حل المسائل الحاسوبية سنفرد فقرة مستقله نتناول فيها انواع خرائط سير العمليات بشكل عام و خرائط سير العمليات المتعلقة بالبرامج بشكل خاص .

الخطوة الرابعة : كتابة البرنامج باحدى لغات البرمجة .

هذه الخطوة في الواقع تعتبر مرحلة مباشرة طالما المخطط التدفقي الانسيابي على مستوى عال من الدقة والتفصيل ، إذ يتم ترجمة الخوارزمية والمخطط التدفقي باستخدام احدى لغات تحرير البرامج، والتي يوجد العديد منها ، إلا أن السؤال الهام الذي يطرح نفسه : أي لغة برمجة يلزم استخدامها ؟

إن الإجابة على هذا السؤال ليس من السهولة كما يبدو للوهلة الاولى ولكن هناك العديد من الأسئلة التي تطرح في هذا المجال ، ومن خلال الاجابات عليها نستطيع تحديد اللغة المناسبة ، ونلخص هذه الاسئلة فيما يلي :

1. هل المبرمج لديه معرفة ودراية بهذه اللغة ؟ واذا لم يكن لديه الدراية فهل يمكنه تعلمها بسهولة ؟
2. هل استخدام هذه اللغة سيحقق أداءا جيدا لهذه النوعية من التطبيقات ؟
3. هل هناك مترجم كفؤ لهذه اللغة ..؟ اذ أنه هناك فرق بين اللغة ومترجمها ، فاللغة هي مجموعة من القواعد والاصطلاحات تستخدم لتمثيل المعلومات من الانسان الى الحاسوب ، آما مترجم اللغة فهو عبارة عن برنامج كتب من قبل مبرمجين ، وظيفته تحويل البرنامج المكتوب بلغة معينة الى لغة الآلة .

(*) اول من استخدم المخططات الانسيابية بصفة رسمية هو جون فون نيومان عام 1945 ، وارتبطت منذ ظهورها بمجالات الحاسوب وما زال استخدامها مستمرا حتى الآن .

4. كيف يتم تشغيل التطبيق المطلوب ؟ إذ كان التطبيق سيتم تشغيله بصفة دورية ولمدة طويلة، عندئذ فان وقت التشغيل الأقل و حيز التخزين الاقل اكثر أهمية من ارتفاع تكاليف اعداد و تجهيز البرنامج التطبيقي و لذلك يفضل استخدام لغة التجميع Assembler مثلا، من جهة اخرى اذا كان وقت استخدام التطبيق لفترة محدودة فانه من الأفضل أن يكون وقت تحرير البرامج أقل ما يمكن وفي هذه الحالة يفضل استخدام احدى لغات المستوى العالي السهلة .

5. هل سيتم تغيير وتعديل البرنامج دوريا ؟ إذ أن اللغات عالية المستوى أسهل في التعديل من اللغات ذات المستوى المنخفض وذلك لقربها من لغة الإنسان .

6. هل من المتوقع تغيير الأجهزة أو نظام التشغيل التي يعمل عليها البرنامج في المستقبل القريب ؟ إذ أنه ليست جميع اللغات متاح لها مترجمات لجميع اجهزة الحواسيب و نظم التشغيل ، كما أن هناك لغات يسهل تحويل البرامج المكتوبة بها من حاسوب الى آخر و من جيل حواسيب الى جيل آخر ..

وقد تعرضنا في الفصل السابق الى أنواع لغات البرمجة الحاسوبية وأهمية كل لغة ، وستتعرض في الفصل القادم الى احدى لغات البرمجة بشكل مفصل (لغة باسكال) ، يدعى البرنامج المكتوب باحدى لغات البرمجة بالبرنامج المصدري Source Program.

(8 – 1– 2) المرحلة الثانية دور الحاسوب :

ونقصد بدور الحاسوب استخدم التقنيات الحاسوبية المختلفة ، وتمر هذه المرحلة بعدة خطوات :

الخطوة الأولى : إدخال البرنامج إلى الحاسوب :

بعد الانتهاء من اعداد البرنامج ، فإنه يتعين إدخاله إلى الحاسوب للتأكد من صحة كتابته من جهة ، ثم لترجمته الى لغة الالة من جهة اخرى ، ويدعى البرنامج الجديد المترجم بالبرنامج الهدف Object Program وتتم الترجمة بواسطة برامج

خاصة تعدها الشركات المنتجة للحواسيب لترجمة البرامج المكتوبة بإحدى اللغات العالية المستوى الى لغة الآلة بهدف تحقيق مهمتين اساسيتين :

1. اكتشاف الاخطاء المعجمية و النحوية (وبعض الاخطاء المنطقية) .
2. ترجمة البرنامج الأولي (برنامج المصدر) المكتوب بإحدى اللغات العالية المستوى والخالي من الأخطاء إلى برنامج بلغة الآلة الذي يدعى بالبرنامج الهدف.

الخطوة الثانية : اختبار البرنامج وتصحيح اخطائه

Testing and Debugging

بعد الحصول على برنامج الهدف ،يتم تجربته للتأكد من صحته منطقيا وذلك بتنفيذه على عينة من البيانات الاختيارية (Test Data) فإذا ما ثبت صحة البرنامج منطقيا ، فإن ذلك يعني صحة طريقة الحل المقترحة .

وبشكل عام تنقسم الأخطاء المحتمل وجودها في أي برنامج الى نوعين أساسيين :

1. أخطاء في قواعد اللغة (Syntax Errors) وهي أخطاء يكتشفها الحاسوب مباشرة لدى تشغيل البرنامج و يعطي الحاسوب رسالة خطأ بخصوصها ، ويمكن تصحيحها مباشرة بالعودة الى البرنامج المساعد help التابع للغة المستخدمة للتأكد من صحة استخدام وكتابة تعليمات اللغة .
2. أخطاء في تصميم البرنامج (Logical Errors) وهي أخطاء منطقية تؤدي الى نتائج خاطئة أو بعيدة عن المنطق و هذه الأخطاء قد تحدث رغم عدم وجود أخطاء في قواعد اللغة، ولا يمكن للحاسوب اكتشافها و بالتالي لايعطي أي رسائل خطاء بخصوصها و ينهي الحاسوب البرنامج نهاية طبيعية ، وهذا النوع من الأخطاء خطير جدا و يجب الإعداد الجيد له لاكتشافه وذلك باختبار البرنامج مرات عديدة على بيانات مختلفة تمثل الواقع الفعلي لظروف تشغيل البرنامج .

الخطوة الثالثة :اعداد البيانات والمعطيات وتنفيذ البرنامج Program Execution

حيث يتم في هذه الخطوة اعداد البيانات و المعطيات الحقيقية عن المشكلة المدروسة ليصار الى ادخالها حسب اسلوب الادخال المتبع في تصميم البرنامج ومن ثم معالجتها حسب تعليمات البرنامج الحاسوبي .

الخطوة الرابعة : الحصول على النتائج Results

في هذه الخطوة تقوم وحدات المخرجات (الطابعة ، الشاشة ،...) بعرض النتائج ليصار الى دراستها و تحليلها ...

(8 – 1 – 3) المرحلة الثالثة دور الانسان : (التوثيق)

دور الإنسان في هذه المرحلة يتمثل بالمبرمج ومحلل النظم ، وتدعى هذه المرحلة بمرحلة توثيق البرنامج (Documentation) وهي عملية تجميع وتنظيم كافة التفاصيل و المعلومات المتعلقة بالبرنامج و كتابتها أو حفظها بأية وسيلة حفظ دائمة مثل الأقراص، الشرائط الممغنطة بحيث يمكن قراءة البيانات و المعلومات الموجودة بها عند الحاجة .

وتشمل هذه التفاصيل و المعلومات تسجيل الآتي :

1. تعريف المشكلة :ومن يستطيع الاستفادة من البرنامج وفي أي مجال يمكن استخدامه ، وما هي الأهداف التي يمكن تحقيقها منه ، و ...
2. وصف النظام :أي تحديد سمات المشكلة و نوع المدخلات المراد استخدامها وشكل ونوع المخرجات المطلوبة بكل وضوح ودقة .
3. وصف البرنامج : و يشمل خوارزمية الحل و خرائط سير العمليات و العينة الاختبارية للبيانات و نتائجها .
4. تعليمات التشغيل :وتشمل تحديد كيفية تحميل البرنامج و تشغيله و ايقافه بالتفاصيل ، و في مختلف ظروف التشغيل .

5. قائمة قواعد الأخطاء : و تشمل تعريف الاجراءات التي يتخذهـــا مســتخدم الحاسوب في حالة حدوث أخطاء معينة أو توقف جزء معين من البرنامج .

(8 – 2) مخططات سير العمليات (المخططات الانسيابية) Flowcharts

(8 – 2 – 1) مفهوم و أهمية مخططات سير العمليات :

كما ذكرنا سابقا إن الخوارزمية هي عبارة عن مجموعـــة مـــن الخطـــوات المتسلسلة التي تصف بصورة مضبوطة وبدون أي غموض جميع الخطوات الرياضيـــة و المنطقية اللازمة لحل مسالة ما، الا أنه في كثير من الاحيان يكون هذا الوصـــف معقدا و صعبا من حيث الالمام بجوانب المسالة و تتبع مراحل حلها، لذلـــك فـــان خرائط سير العمليات تمثل وصفا تصويريا لخطوات الخوارزمية، وكثير من المـــبرمجين يفضلون خريطة سير العمليات عن الخوارزميات، اذ يمكن بواسطة الخريطة تتبـــع التسلسل المنطقي لحل المسالة بسهولة، وفي بعض الاحيان تكـــون عمليـــة كتابـــة الخوارزميات اسهل بالاعتماد على خرائط سير العمليات من كتابـــة الخوارزميـــة مباشرة .

ولخرائط سير العمليات (المخططات الانسيابية) أهمية كبيرة نوجزها بما يلي:

1- تعطي تصور كامل عن خطوات حل المسألة في ذهن المـــبرمج تمكنـــه مـــن الإحاطة الكاملة بكل أجزاء المسألة من بدايتها وحتى نهايتها .

2- تساعد المبرمج على تشخيص الأخطاء التي تقع عـــادة في الـــبرامج وبخاصـــة الأخطاء المنطقية منها ، والتي يعتمد اكتشافها على اسلوب التسلسل المنطقـــي لخطوات حل المسألة لدى المبرمج .

3- تساعد المبرمج في ادخال أي تعديلات في أي جزء من أجزاء المسألة بســـرعة ودون الحاجة لأعادة دراسة المسألة برمتها من جديد .

4- تساعد المبرمج على تتبع التسلسل المنطقي لخطوات الحل وخاصة في المســـائل التي تكثر فيها الاحتمالات والتفرعات .

5- تعتبر رسوم خرائط سير العمليات المستعملة في تصميم حلول بعض المسائل مرجعا في حل مسائل أخرى مشابه ومفتاحا لحل مسائل جديدة لها علاقة مع المسائل القديمة المحلولة .

(8-2-2) الاشكال والرموز الاصطلاحية المستخدمة في مخططات سير العمليات:

هناك رموز واصطلاحات متعارف عليها دوليا تستخدم عند اعداد مخطط سير العمليات لمشكلة معينة ، هذه الرموز والاصطلاحات جاءت نتيجة جهود المعهد القياسي الامريكي (American National Standard Institute ANSI) من هذه الرموز

تستخدم للبداية Start ، وللنهاية End

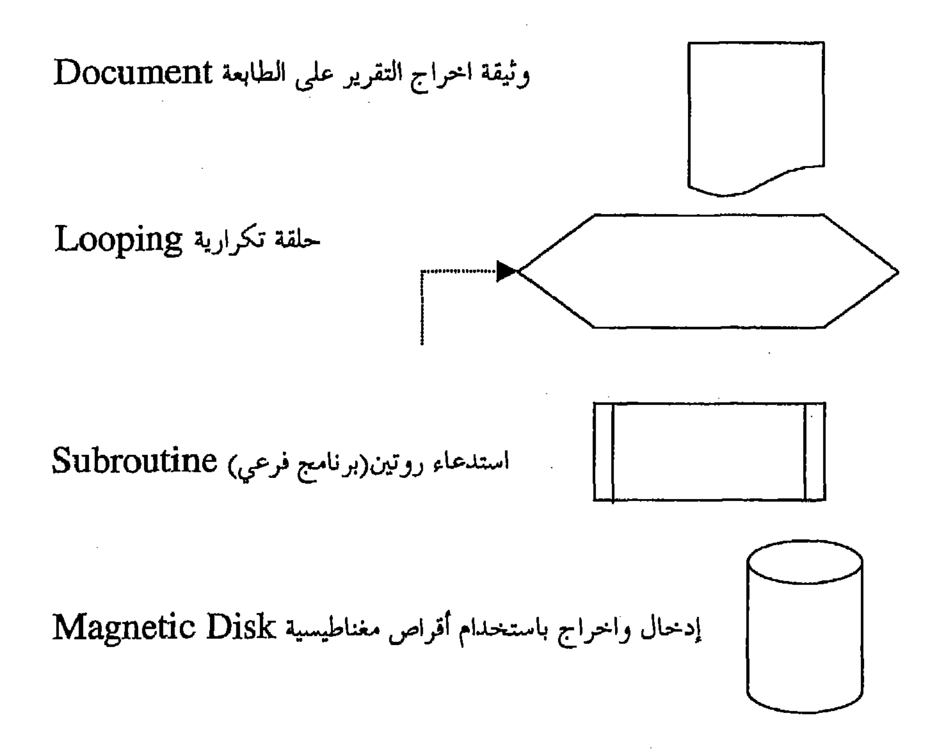
للادخال Input والاخراج Output
دون تحديد وسيلة الادخال والاخراج

عملية حسابية Calculation

اتخاذ قرار منطقي (اختبار مقارنة)
Logical Decision Making

أسهم بين الاشكال والرموز الاصطلاحية تدل على اتجاه سير العمليات في خريطة سير العمليات flow lines

نقطة توصيل وربط بين أجزاء سير العمليات Connection



بالإضافة إلى رموز أخرى لا ضرورة لذكرها بسبب استخدامها القليل جدا .

(8—2—3) انواع المخططات الانسيابية :

تنقسم المخططات الانسيابية (خرائط سير العمليات) بشكل عام الى قسمين رئيسيين هما :

1) المخططات الانسيابية للأنظمة Systems Flowcharts

وهذا النوع يحدد بالرسوم و العلاقات المنطقية التي تربط بين مجموعة العمليات المتتالية و المترابطة للنظام المطلوب ويتكون من عدد من الخرائط المهمة كل منها توضح وظيفة أو مرحلة معينة داخل النظام من خلال النقاط التالية :

1. سير البيانات من حيث طريقة نقلها وتغذية الحاسوب بها وكذلك تناولها ومعالجتها.

2. طريقة ربط برامج النظام بعضها ببعض .

3. العلاقة بين الدوائر المستخدمة للحاسوب و النظام المقترح .

4. تحديد شكل وطبيعة النتائج الخاصة بكل برنامج .

وهناك أمثلة كثيرة على خرائط النظم منها ما تستخدمه الشـــركات والمؤسســـات لوصف انظمتها المختلفة.

2) المخططات الانسيابية للبرامج Programs Flowcharts

هذا النوع يحدد بالرسوم العلاقات المنطقية و التسلسل العام بين مجموعـــة من الاحداث والعمليات المترابطة والمتتالية التي يتكون منها البرنامج .

و هذا النوع من المخططات له أهمية بالغـــة في كتابـــة الـــبرامج الحاســـوبية ، وسنتناولها في الفقرات اللاحقة بالتفصيل .

(8–3) انواع المخططات الانسيابية للبرامج الحاسوبية :

يمكن تصنيف المخططات المتعلقة بالبرامج الى ثلاثة انواع :

1 – مخططات التتابع البسيط Simple Squential Flowcharts

2 – مخططات التفرع Branched Flowcharts

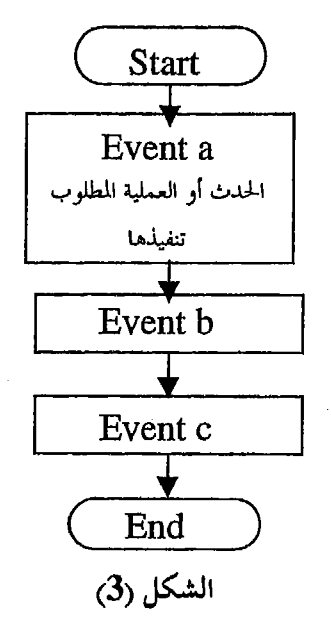
3 – مخططات التكرار Loop Flowcharts

ويمكن للبرنامج الواحد ان يشمل على أكثر من نوع واحد من هذه الانواع .

(8 – 3 – 1) مخططات التتابع البسيط

Simple Sequential Flowcharts

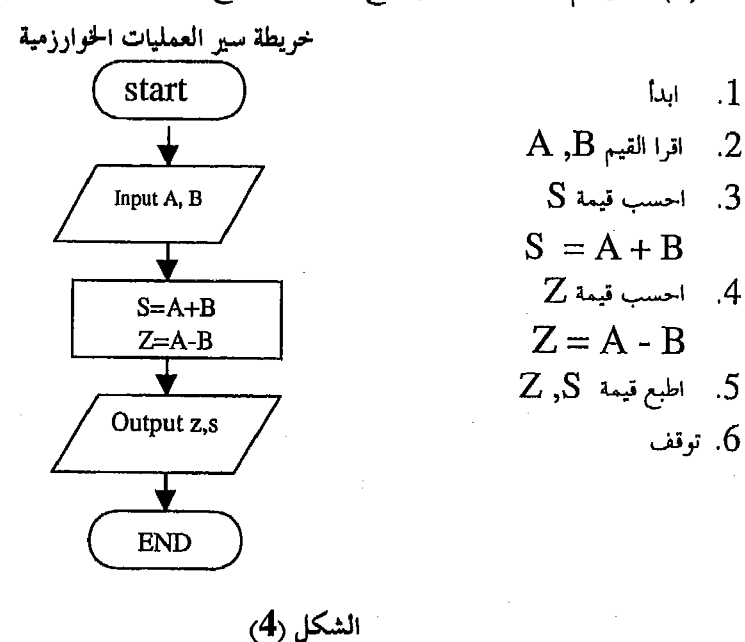
هذا النوع من المخططات يخلو من التفرعات والتكرارات او أي اختبـــار منطقي والشكل العام للمخطط الانسيابي كما في الشكل (3)



الشكل (3)

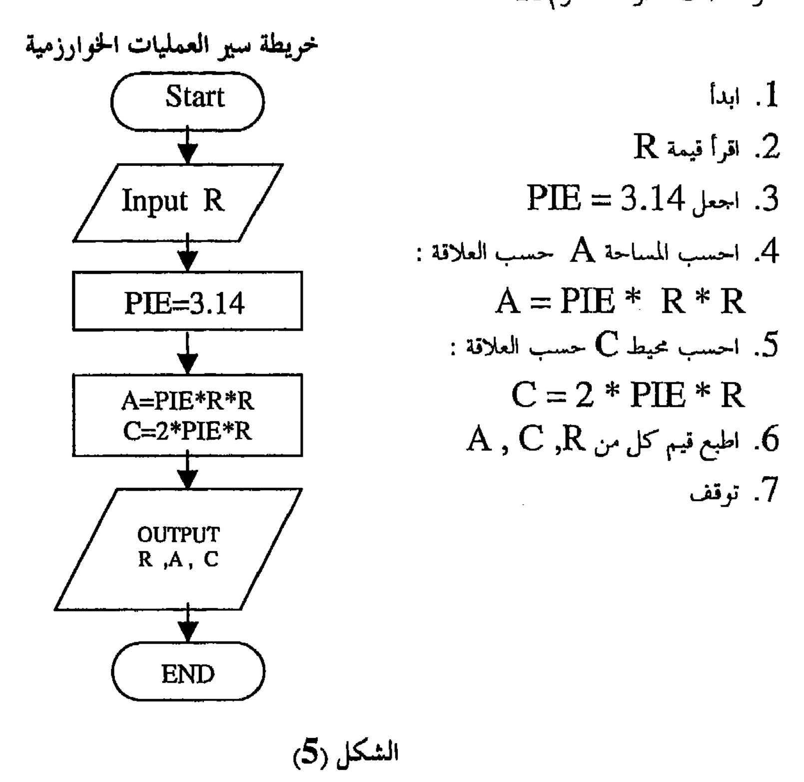
حيث : Event الواردة في الشكل السابق تعني الحـــدث او العمليـــة المطلـــوب تنفيذها إما عملية حسابية أو عملية إدخال أو إخراج .

مثال (4) : ارسم مخطط سير برنامج لايجاد مجموع عددين ما وحاصل طرحها .



الشكل (4)

مثال (5) : اكتب خوارزمية وارسم مخطط سير العمليات لبرنامج حساب مساحة ومحيط دائرة نصف قطرها معلوم R ؟



الشكل (5)

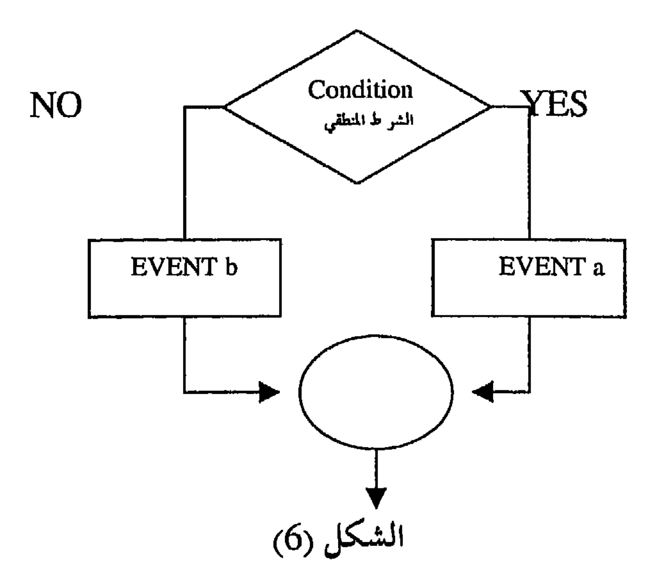
(8 3 2) مخططات التفرع (الاختيار)

يتضمن هذا النوع من المخططات اتخاذ قرار أو مفاضلة بين خيارين أو اكثر وهناك شكلين لمخططات الاختيار :

<u>الشكل الاول</u> : انظر الشكل (6)

في هذا الشكل اذا كان جواب الشرط المنطقي (Condition) نعم (Yes) فان الحدث التالي في التنفيذ هو (a)، اما اذا كان جواب الشرط بلا (No)فان الحدث التالي في التنفيذ هو (b) .

ويعرف هذا الشكل بصيغة اذا المزدوجة أي IF - THEN - ELES

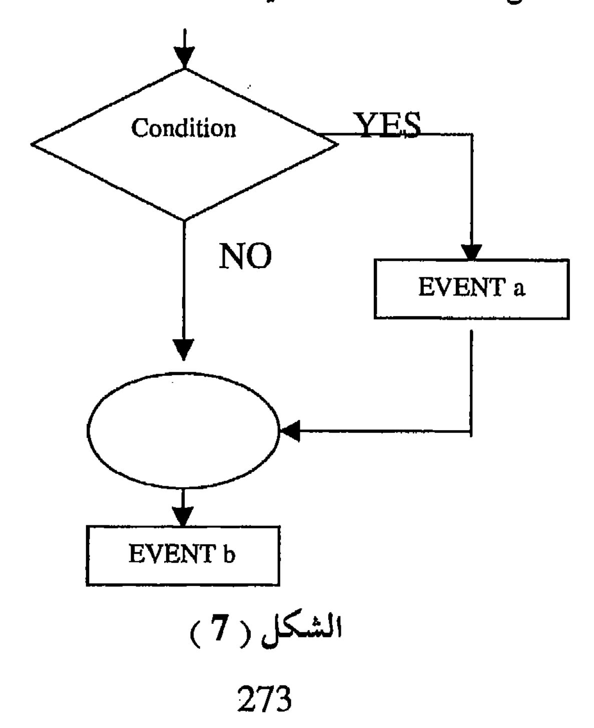


الشكل (6)

<u>**الشكل الثاني**</u> : انظرالشكل (7)

في هذا الشكل إذا كان جواب الشرط المنطقي (Condition) نعـــم (Yes) فان الحدث التالي في التنفيذ هو (a)، أما إذا كان جواب الشرط بلا (No)فـــان الحدث التالي في التنفيذ هو (b) .

ويعرف هذا الشكل بصيغة اذا البسيطة أي IF - THEN - STATEMENT



الشكل (7)

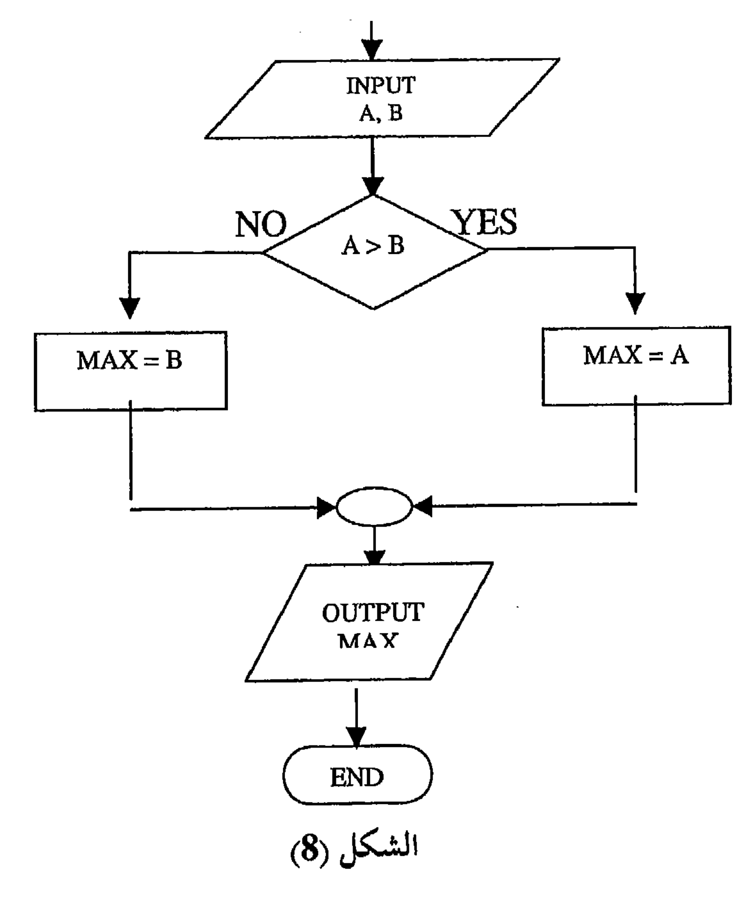
مثال (6) :اكتب خوارزمية الحل و ارسم مخطط سير العمليات لقـــــراءة قيمتـــين معلومتين واختيار اكبرهما باستخدام الشكل الاول و الشكل الثاني .

الخوارزمية باستخدام الشكل الاول :

1. ابدأ
2. أدخل قيمة كل من A , B
3. اذا كانت A اكبر من B : عندئذ اجعل : Max = A

   والا اجعل : Max = B
4. اطبع النتيجة Max
5. توقف

والشكل (8) يبين خريطة سير العمليات لهذه الخوارزمية .



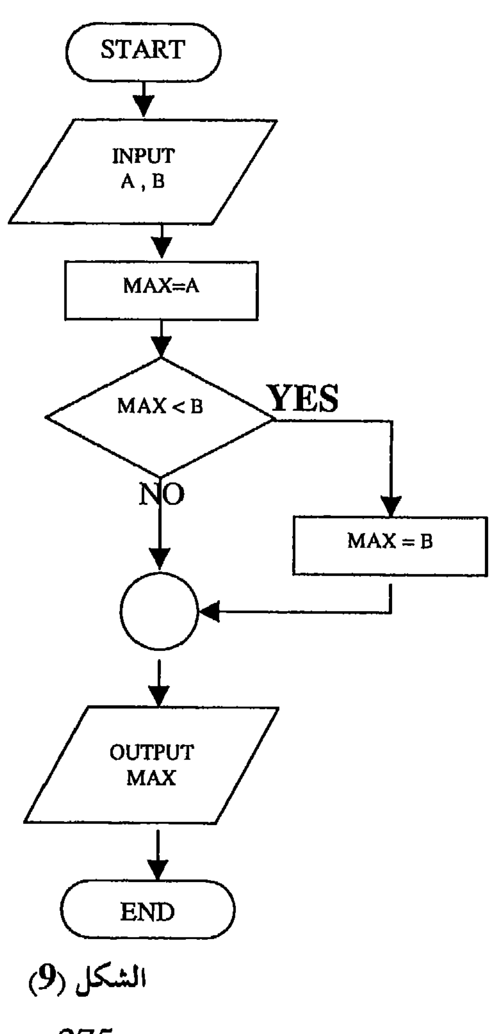
الشكل (8)

باستخدام الشكل الثاني :

الخوارزمية :

1. ابدأ
2. ادخل قيمة كل من A , B
3. اجعل Max = A
4. اذا كانت Max < B عندئذ اجعل Max = B
5. اطبع Max
6. توقف

والشكل (9) يبين سير العمليات لهذه الخوارزمية .



الشكل (9)

مثال (7) : اكتب خوارزمية وارسم مخطط سير العمليات لاعادة ترتيب قيمتين ترتيبا تصاعديا.

الخوارزمية :

1. ابدأ
2. ادخل القيمتين A , B
3. اذا كانت A > B عندئذ استبدل بالشكل التالي :

   اجعل Z = A

   اجعل A = B

   اجعل B = Z
4. اطبع A , B
5. توقف

والشكل (10) يبين خريطة سير العمليات لهذه الخوارزمية .

ملاحظة :بالإضافة الى الشكلين (9) و (10) لمخططات الاختيار هناك مخططات يمكن ان تحتوي الشكلين معا و تدعى مخططات اذا المتداخلة Nested IF Statement والمثالين التاليين يوضحان ذلك :

مثال (8) : اكتب خوارزمية الحل و ارسم مخطط سير العمليات للمشكلة التالية :

يقدم احد المخازن خصما على مبيعاته حسب ما يلي :

1- 20% اذا كانت طلبية الزبون اكثر من 200 دينار
2- 15% اذا كانت طلبية الزبون بين 100 دينار واقل من 200 دينار
3- بدون خصم على الطلبية التي قيمتها أقل من 100 دينار

الخوارزمية :

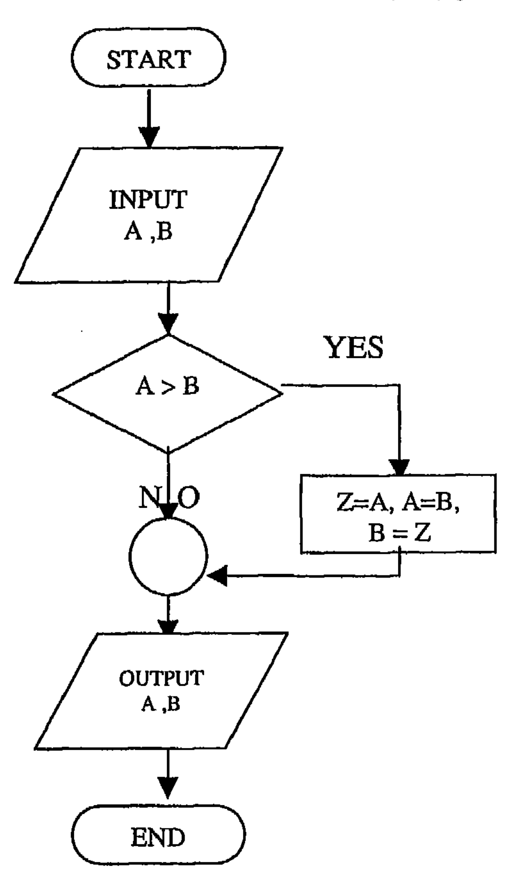
1. ابدأ
2. أدخل قيمة طلبية الزبون S
3. اذا كانت S > 200 احسب قيمة D حسب العلاقة التالية :

$$D = S * 0.20$$

والا اذا كانت $100 > S > 200$ احسب قيمة D حسب العلاقة التالية:

$$D = S * 0.15$$

و إلا فإن $D = 0$



الشكل (10)

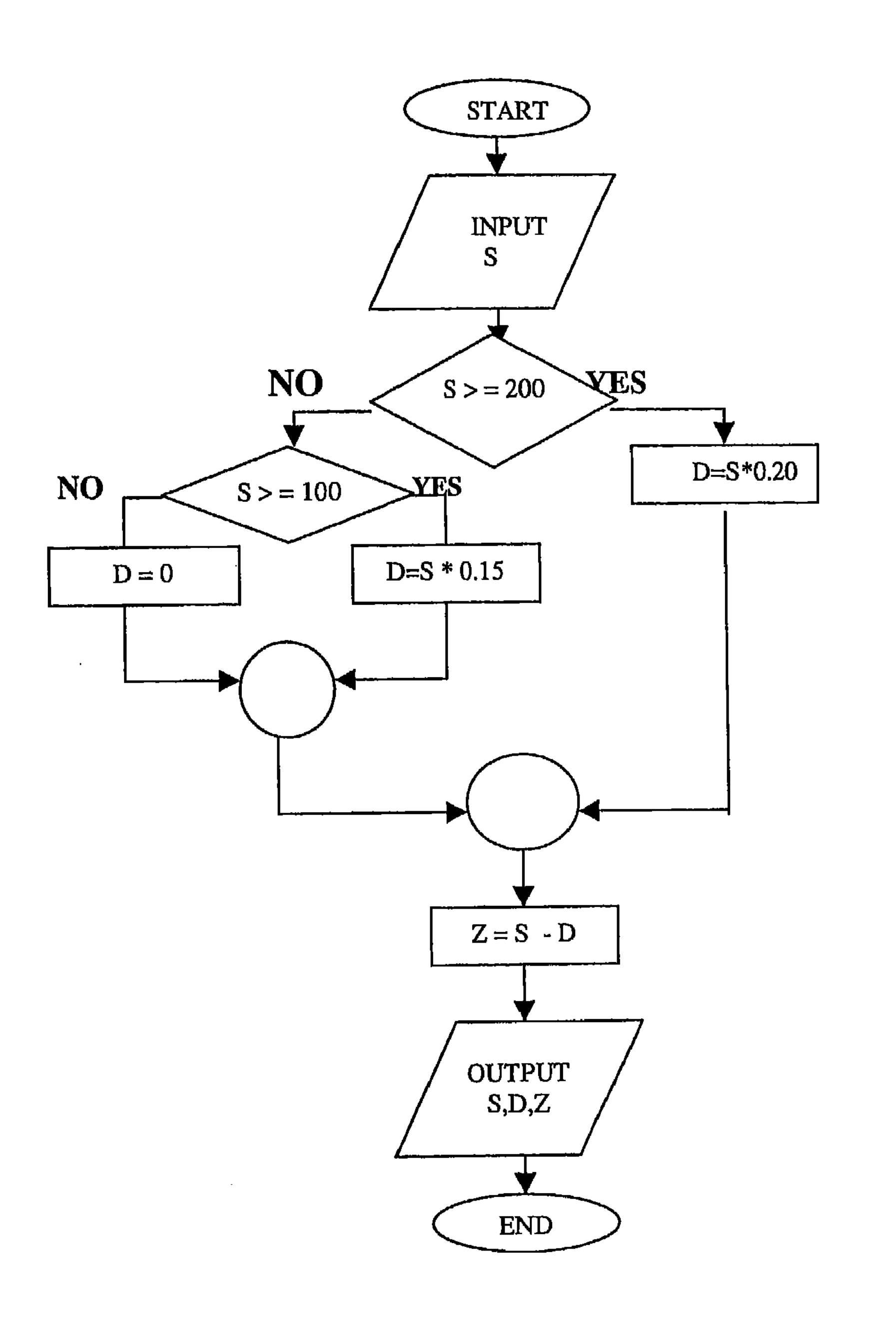
4. قيمة الطلبية النهائي :

$$Z = S - D$$

5. اطبع قيمة Z , D , S

6. توقف

والشكل (11) يبين مخطط سير العمليات لهذه الخوارزمية .


START
INPUT
S
NO
S > = 200
YES
NO
S > = 100
YES
D=S*0.20
D = 0
D=S * 0.15
Z = S - D
OUTPUT
S,D,Z
END

الشكل (11)

مثال (9) ارسم مخطط سير العمليات وخوارزمية الحل لحساب الدالة F (X , Y) المعرفة كما يلي :

$$F(X,Y) = \begin{cases} X+Y & ; \quad X>Y \\ X^2+5 & ; \quad X=Y \\ Y+2X & ; \quad X<Y \end{cases}$$

حيث قيم Y , X معلومتين

الخوارزمية :

1- ابدأ

2- إذا كانت X > Y احسب :

$$F = X + Y$$

اما اذا كانت X = Y عندئذ احسب العلاقة :

$$F = X*X + 5$$

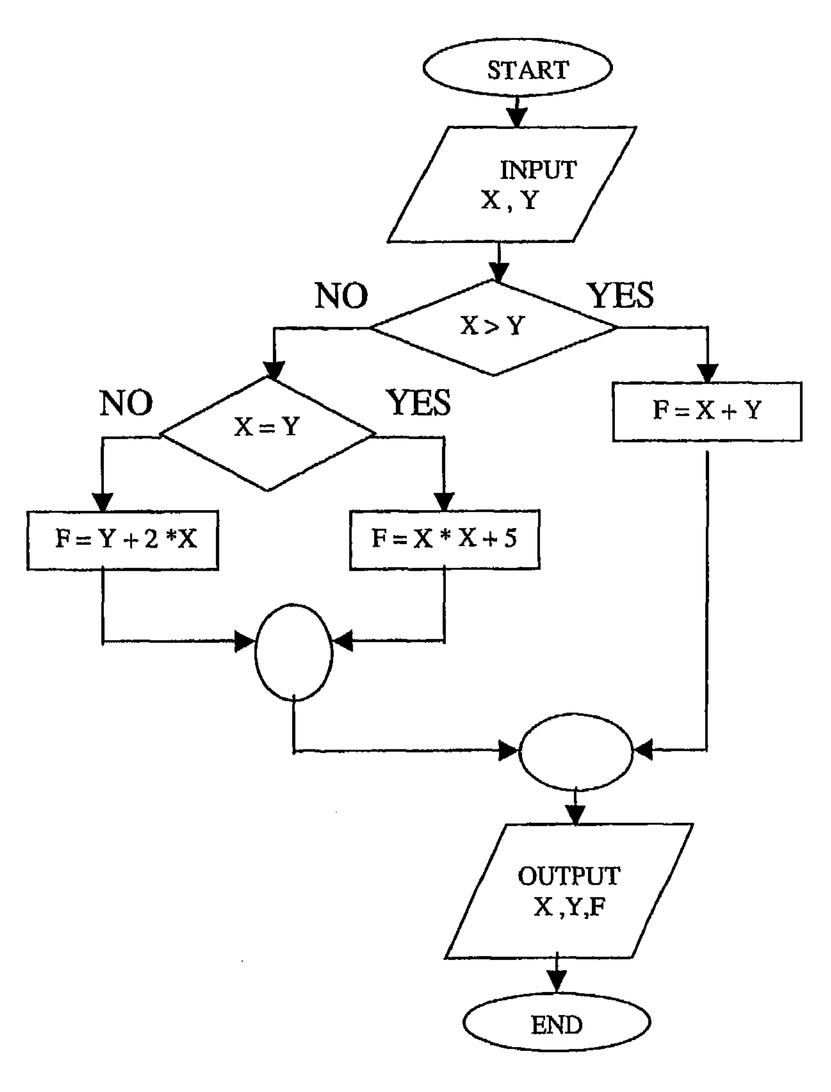
و إلا فاحسب :

$$F = Y + 2*X$$

3- اطبع قيمة كل من F , Y , X

4- توقف

والشكل (12) يوضح مخطط سير العمليات لهذه الخوارزمية .

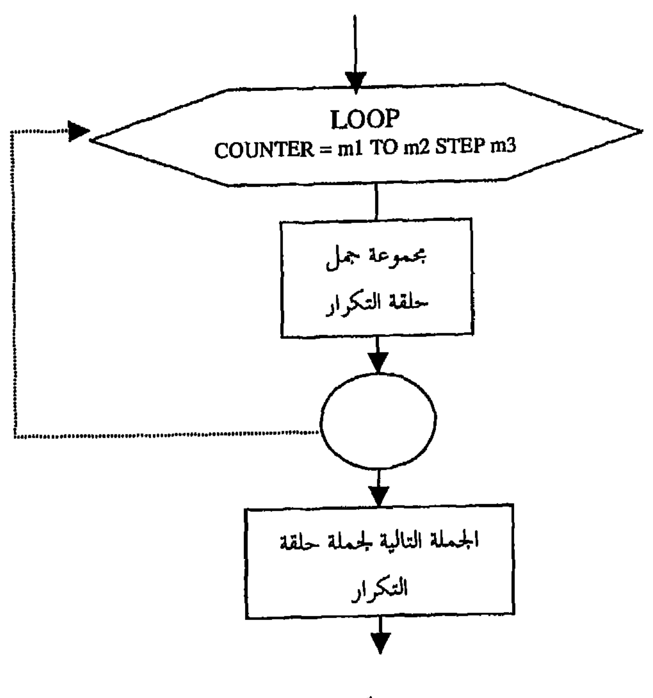


الشكل (12)

(8 3 3) مخططات التكرار :

و ينقسم هذا النوع من الخرائط الى نوعين :

1- اما تكرار تنفيذ مجموعة من الجمل عددا محدودا من المرات و يدعى في هـــذه الحالة بالتكرار غير المشروط ويأخذ الشكل العام التالي (انظر الشكل (13))



الشكل (13)

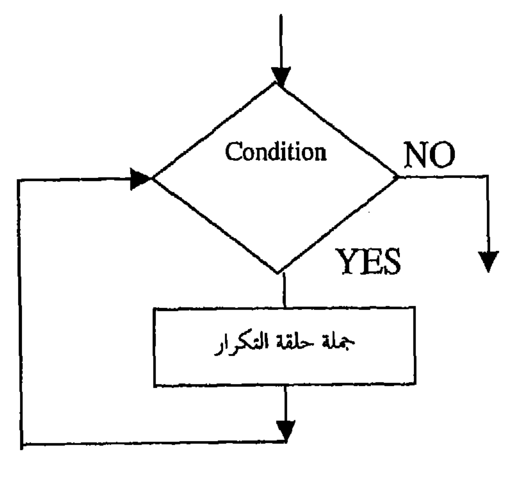
حيث : Counter اسم عداد، m1 القيمة الأولية للعداد ،m2 القيمة النهائية للعداد، m3 مقدار التزايد .

ويتم تكرار مجموعة جمل حلقة التكرار عددا من المرات ابتداءا من القيمة الابتدائية الاولية m1 و حتى القيمة النهائية m2 بزيادة مقدارها m3

2- أو تكرار جمل حلقة التكرار الى ان يتحقق شرط معين ويدعى في هذه الحالة بالتكرار المشروط ويأخذ أحد الشكلين التاليين :

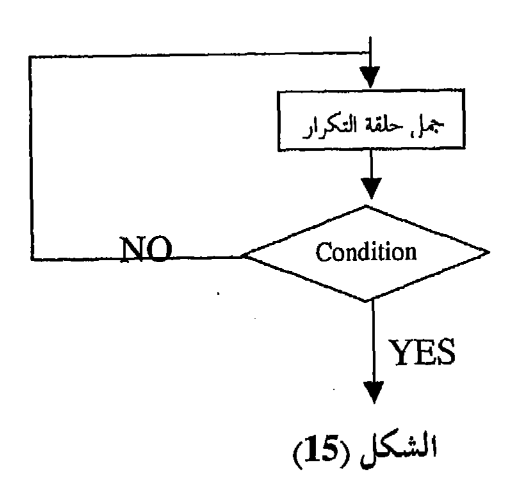
الشكل الأول : الشكل (14) ، في هذه الحالة يتكرر تنفيذ جمل حلقة التكرار ما دام الشرط محققا و تعرف هذه الحالة في اغلب لغات البرمجة بـــ :

WHILE .... DO



الشكل (14)

الشكل الثاني : الشكل (15) ، في هذه الحالة يتكرر تنفيذ جمل حلقة التكـــرار حتى



الشكل (15)

يتحقق الشرط و تعرف هذه الحالة بـــ :

REPEAT ...... UNTIL

و قبل أن نعطي أمثله على هذا النوع من الخرائط لابد من التطرق الى مفــــهومين اساسين يستخدمان بشكل واسع لدى كتابة البرامج ورسم الخرائط وهما مفـــهومي العداد و المجمع .

اولا : مفهوم العداد Counter

قد نحتاج في كثير من الاحيان الى ان نعد ظاهرة معينة عدد من المرات، مثلا اذا طلب ان نعد كل من الطلاب والطالبات في الصف ،ان هـذه العملية بالنسبة للانسان بديهيه وضمن قدراته العقلية التي يكتسبها منذ طفولته ، إلا أن الحاسوب يحتاج الى اسلوب معين او ما يسمى بتصميم خوارزمية للعد Counting Algorithm و التي تتضمن الخطوات التالية :

1. اجعل العداد يبدأ بقيمه ابتدائية تساوي الصفر Counter = 0
2. اجعل قيمه العداد تساوي القيمة السابقة مضافا اليها الزيادة الدورية m :

$$Counter = Counter + m$$

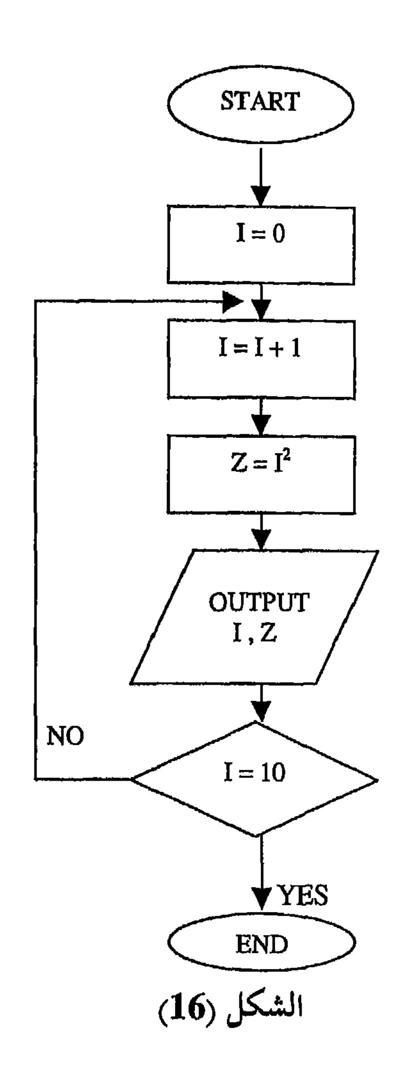
3. نفذ التعليمات والاوامر المراد تكرارها.
1. اعد تكرار تنفيذ الخطوات (2) و (3) عددا من المرات حسب حالة التكرار .

مثال (10) اكتب خوارزمية وارسم خريطة سير العمليات لطباعة الاعداد الطبيعية من 1 الى 10 و مربعاتها :

* الخوارزمية باستخدام التكرار المشروط :

1. ابدأ
2. أجعل I = 0
3. أجعل I = I + 1
4. أحسب Z من العلاقة التالية : Z = I * I
5. اطبع I , Z
6. هل I = 10 ، اذا كان الجواب No ، اعد تكرار الخطوات (3 و 4 و 5 و6)
7. توقف

والشكل (16) يبين مخطط سير العمليات لهذه الحالة .

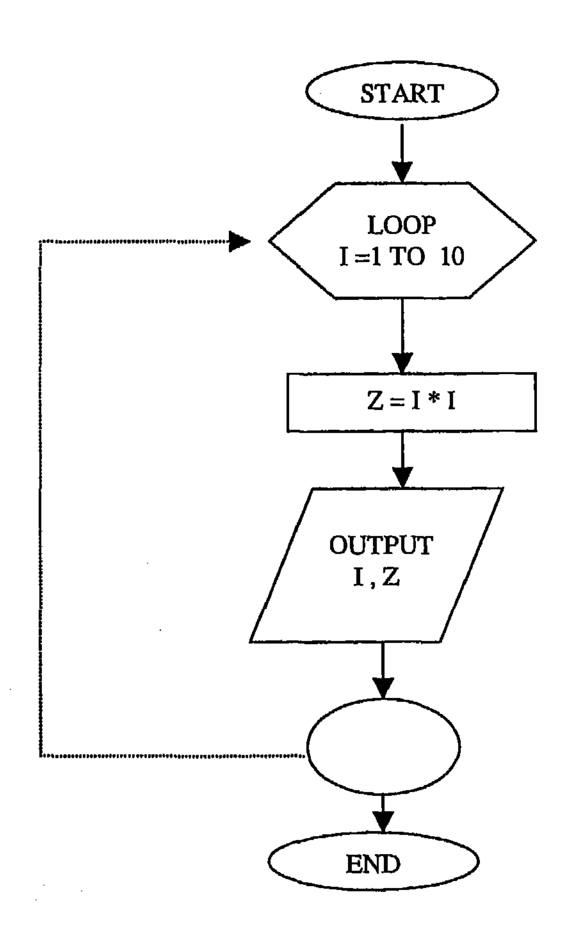


الشكل (16)

- الخوارزمية باستخدام التكرار الغير المشروط :

1. ابدأ
2. كرر مايلي عشرة مرات
   - احسب قيمة Z حيث : $Z = I * I$
   - اطبع قيمة Z , I
3. توقف

والشكل (17) يبين مخطط سير العمليات لهذه الخوارزمية .



الشكل (17)

ثانيا : مفهوم المجمع Summer

قد نحتاج في بعض الاحيان الى جمع بمجموعة كبيرة من الاعداد التي تمثل معطيات ظاهرة معينة ، فمثلا ، كأن ترغب في ايجاد الوسط الحسابي لاعمار طلاب السنة الاولى في كلية الاقتصاد، ولتحقيق هذا يجب ان نحسب بمجموع اعمار الطلاب ،أي اننا بحاجة الى ايجاد مجموع عدد معلوم من القيم ، وهنا نشير الى أنه من الناحية العملية لا يمكن إعطاء رمزا أبجديا لكل عدد يمثل عمر طالب إذ أنه في هذه الحالة نحتاج إلى مئات من الرموز ، و للتغلب على هذه المشكلة نستخدم ما نسمى بخوارزمية التجميع (Summers Algorithm) ، ونحتاج لاستخدام

هذه الخوارزمية متغيرين اثنين فقط إحداهما يمثل المتغير الذي نجمعه والآخـــر يمثـــل المجمع الاجمالي ، و الخوارزمية كما يلي :

1- اجعل قيمة المجمع الأولية تساوي الصفر $Sum = 0$

(ويمكن جعلها أي قيمة حسب المشكلة المدروسة)

2- اقرا القيمة التالية من القائمة ولتكن x

1- اجعل القيمة الجديدة للمجمع تساوي القيمة السابقة له مضافا اليها القيمـة المقروءة أي أن: $Sum = Sum + x$

2- اعد تنفيذ الخطوات (3 , 2) حسب حالة التكرار .

مثال (11) : اكتب خوارزمية و ارسم مخطط سير العمليات لايجاد الوسط الحسابي لاعمار طلاب قسم الحاسوب .

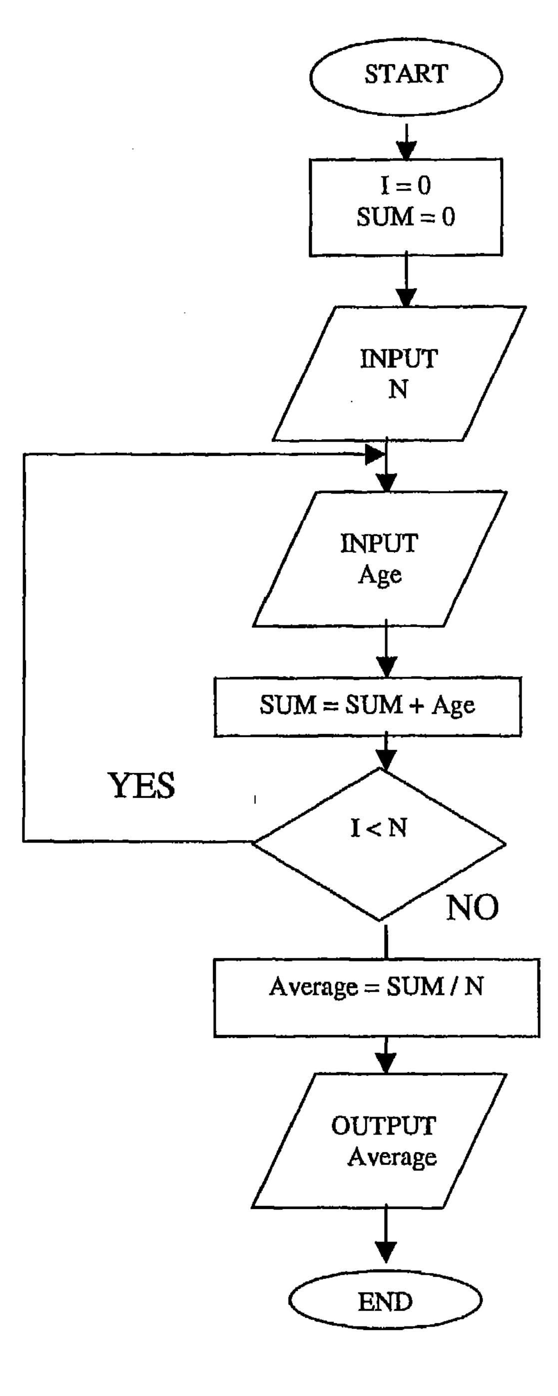
* الخوارزمية باستخدام التكرار المشروط :

1. ابدأ
2. اجعل العداد $I = 0$ و المجمع $Sum = 0$
3. ادخل عدد الطلاب N
4. اضف واحد الى العداد I أي $I = I+1$
5. ادخل عمر الطالب التالي Age
6. اجعل $Sum = Sum + Age$
7. اذا كان $I < N$ عد الى الخطوة (4) والا تابع ،
8. احسب الوسط الحسابي :

$$Average = Sum / N$$

9. اطبع قيمة Average .
10. توقف

والشكل (18) يوضح مخطط سير العمليات لهذه الخوارزمية .


START
I = 0
SUM = 0
INPUT
N
INPUT
Age
SUM = SUM + Age
YES
I < N
NO
Average = SUM / N
OUTPUT
Average
END

الشكل (18)

الخوارزمية باستخدام التكرار غير المشروط :

1- ابدأ .

2- اجعل SUM = 0 .

3- اقرأ عدد الطلاب N.

4- كرر مايلي N مرة .

- إقراء عمر الطالب التالي Age .

- أجعل sum = sum + Age

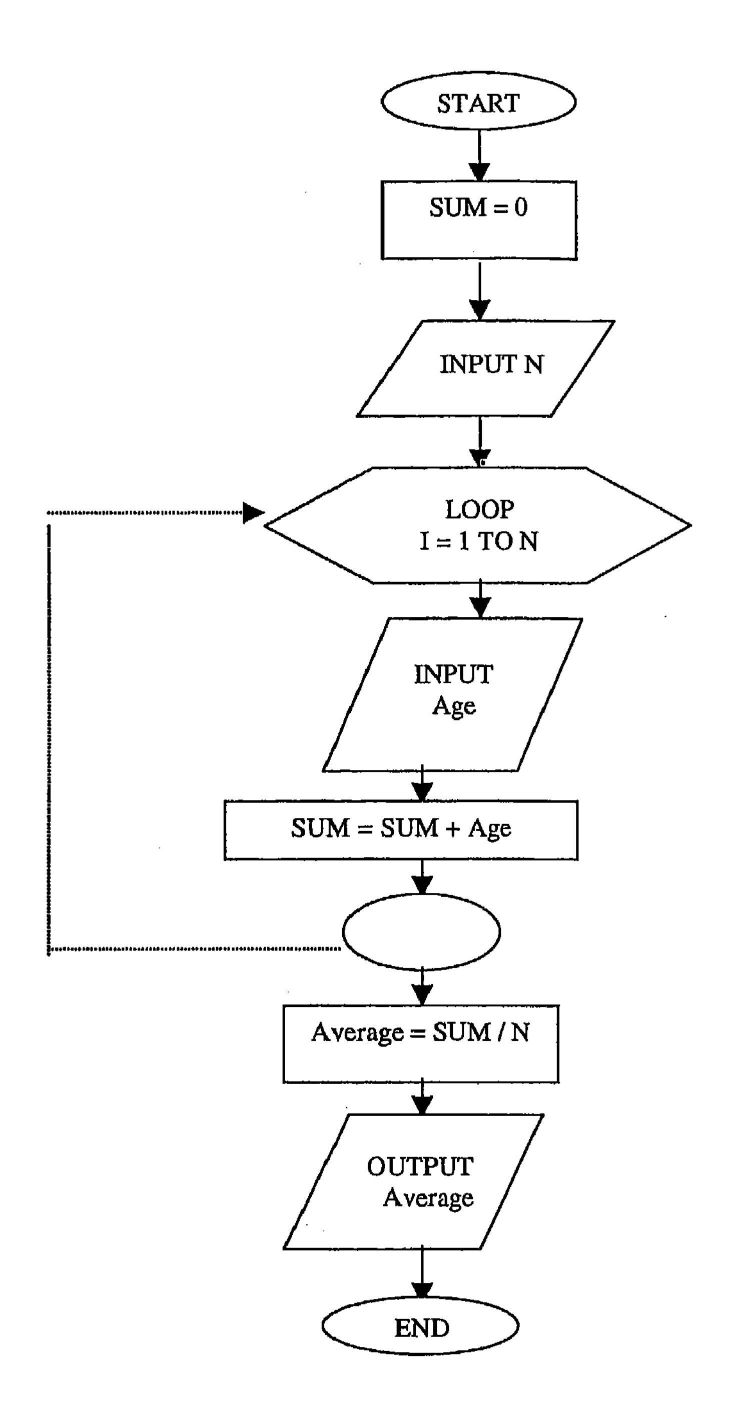
5- أحسب الوسط الحسابي :

Average = sum / n

6- اطبع قيمة Average

7- توقف .

والشكل (19) يوضح مخطط سير العمليات لهذه الخوارزمية .


START
SUM = 0
INPUT N
LOOP
I = 1 TO N
INPUT
Age
SUM = SUM + Age
Average = SUM / N
OUTPUT
Average
END

الشكل (**19**)

تمارين وأسئلة عامة

1) عدد المراحل الأساسية لحل مسألة حاسوبيا ؟

2) عرف ما يلي : الخوارزمية   المخطط الانسيابي لسير العمليات   البرنـــامج   الهدف   البرنامج المصدري .

3) اذكر بعض فوائد مخططات سير العمليات مع الشرح ؟

4) عدد انواع مخططات سير العمليات مع الشرح ؟

5) اكتب خوارزمية الحل و ارسم مخطط سير العمليات لكل من المشاكل التالية :

1 - تحديد العدد الأكبر من بين ثلاثة اعداد A , B , C

2 - تحديد العدد الأصغر من بين ثلاثة اعداد A , B , C

3 - إيجاد جذور معادلة من الدرجة الثانية من الشكل $AX^2+ BX +C =0$

4 - إيجاد مضروب عدد حسب القاعدة التالية :

$$N! = \begin{cases} 1 & ; \quad N = 0 \\ 1*2*3*4*......*N & ; \quad N \quad 0 \end{cases}$$

5 - حساب قيمة الدالة : 5 $Y = 2N2$

لجميع قيم X تقع في المجال [2 ,2-] ويتزايد مقداره 0.5

6 - حل جملة معادلتين بمجهولين .

7 - جمع مصفوفتين .

8 - ضرب مصفوفتين .

9 - حل جملة معادلات باستخدام طريقة تكرار .

الفصل التاسع

البرمجة بلغة باسكال

PASCAL PROGRAMMING

(اساسيات ومكونات — جمل الإدخال والإخراج — جمل التحكم والتكرار)

(9 — 1) المقدمة

تتصف لغة Pascal بأنها ذات أغراض متعددة ومختلفة في مجالات عـــدة، هندسية، وتجارية ، وعلمية ، لهذا فقد اخترناها لتكون اللغة المفضلة للجميع وهـــذه اللغة قد تم تطويرها اساساً من لغة Algol والتي وضعت عام 1960 وسميـــت في حينه البرمجة البنائية، وفي أوائل السبعينات طورها العالم (نيكلوس ويرث) وهو مـــن جامعة زيوريخ للتكنولوجيا سويسرا- وسماها بلغة Pascal نســـبة إلى العـــالم الفرنسي Pascal الذي أخترع أول حاسبة ميكانيكية في منتصف القرن الســـادس عشرة. إن لغة باسكال تطورت تطوراً سريعاً وأصبح لها عدة أشكال قياسية منـــها ISO /ANS1 والتي تبنتها مؤسسة المواصفات الدولية وكذلك معهد المواصفـات الوطني الأمريكي ، ثم بعد ذلك تطورت هذه اللغة وانتشر اســـتخدامها انتشـــاراً واسعاً وبدأ استخدام النوع المطور منها والمسمى Turbo-Pascal ، إن الســـبب في انتشار وتطور هذه اللغة يعود إلى سهولتها وقوتها البنائية ولاستخداماتها المتعـددة وملائمتها للاستخدام في مختلف أنواع الحواسيب وكذلك شمولها بمجموعـــة مـــن اللغات الأقدم منها مثل Algol,Cobol,Basic,Fortran وســـرعة إعطائـــها النتائج قياساً إلى اللغات الأخرى بالإضافة إلى كونها لغة تدريبية عالية المستوى .

هذا بالإضافة إلى ان هذه اللغة ركزت تركيزاً مباشراً على مفهوم البرمجة الهيكلية Structured – Programming والتي تسمى في بعض الأحيان البرمجة من الأعلى إلى الأسفل Top – Down Programming اي لغة التركيب المنطقي المتسلسل . لهذا فإن برنامج لغة باسكال يمكن أن يتكون من الأجزاء التالية:

أ. **الجزء الأول** : وهو الجزء المسؤول عن كتابة وذكر أسم البرنامج والذي يأتي بعد كلمة PROGRAM ومن الممكن أن يتكون من سطر واحد أو أكثر.

ب. **الجزء الثاني** : وهو الجزء المسؤول عن التوضيحات والتعريفات للبرنامج وهو يحوي جملة واحدة أو اكثر من جمل التوضيح والإعلان عن جميع العناوين والثوابت والمتغيرات والإجراءات والإقترانات في البرنامج .

جـ. **الجزء الثالث** : وهو الجزء الفعّال والمهم في البرنامج والذي يضم جمل باسكال الحسابية والمنطقية والذي يعتبر الجزء الرئيسي لبرنامج Pascal .

ويمكن من خلال برنامج باسكال البسيط التالي توضيح الأجزاء المذكورة اعلاه وعلى الشكل التالي :

```
PROGRAM    Prog1(Input,Output) ;                         الجزء الأول
   (* Program to Compute and to
        Print the sum of two given number *)

VAR                                                       الجزء الثاني
        a,b, Sum, : INTEGER ;

BEGIN                                                     الجزء الثالث
        WRITE ( 'Enter the value of a :' ) ;
        READ (a) ;
        WRITE ('Enter the Value of b :') ;
        READ (b) ;
        Sum := a+b ;
        WRITELN ( 'Sum = ', sum ) ;
END.
```

وقبل الخوض في تفاصيل البرمجة الهيكلية وتعليماتها ومكوناتها لا بد من التعرف على أساسيات هذه اللغة .

(9 – 2) أساسيات ومكونات لغة باسكال

Pascal Fundamentals and Symbols

إن لغة باسكال تتكون من مجموعة من الرمـــوز والأرقـــام والمســـميات والكلمات والأسماء التعريفية ، ويمكن تقسيم لغة Pascal إلى :

(9–2–1) مجموعة رموز باسكال Pascal Character Set

إن مجموعة رموز لغة باسكال تنقسم إلى ثلاث أنواع هي :

أ. مجموعة الأرقام العشرية (Numeric digits) والتي تسمى في كثـــير مـــن الأحيان أرقام اللغة العربية العشرية وهي عشرة أرقام هي :

0,1,2,3,4,5,6,7,8,9

ب. مجموعة الحروف اللاتينية أو (أحرف اللغة الإنجليزية) والتي تتكون مـــن 26 حرفاً وهي (Letters)

A, B, C, D, E, F, G, H, M, N, L, P, Q, R, ...

بنوعيها الحروف الكبيرة (Capital Letters) والحروف الصغيرة (**Small Letters**)

جـ. مجموعة الرموز الخاصة (Special Characters) والتي تشمل العمليات الأربع والأشارات المنطقية وبعض الاشارات والعلامات الخاصة كما يلي :

)	>	,	+
(	>=	:	–
] أو .)	>		*
[أو (.	<=	;	/
} أو *)	<>	/	:=
{ أو (*		& أو @	=

(9-2-2) الكلمات المحجوزة (Reserved Words)

هذه الكلمات حوالي 39 كلمة تستخدمها لغة باسكال والتي تعتبر كلمات محجوزة قياسية معروفة مسبقاً لدى الحاسوب وتكتب هذه الكلمات عادة بحروف انكليزية كبيرة ولها معان خاصة بها تؤديها في برنامج باسكال ، ولا يجوز للمبرمج الذي يقوم بكتابة البرنامج بلغة الباسكال من إعادة تعريف هذه الكلمات أو استخدامها لغرض آخر غير المخصص لها ، ولهذا سميت كلمات محجوزة وهي كما يلي :

AND	FUNCTION	PACKED	VAR
ARRAY	GOTO	PROCEDURE	WHILE
BEGIN	IF	PRIGRAM	WITH
CASE	IN	RECORD	CONST
LABEL	REPEAT	DIV	MOD
SET	DO	MODULE	STRING
DOWNTO	NIL	THEN	ELSE
NOT	TO	END	OF
TYPE	FILE	OR	UNTIL
FOR	OTHERWISE	VALUE	

(9-2-3) مسميات المبرمج التعريفية (الأسماء التعريفية)Identifiers

وهو الاسم الرمزي للمتغير وهو أيضاً عبارة عن اسم لموضع تخزيني يعطى في البرنامج لتعريف قيمة الثابت أو المتغير أو الاجراء أو البرنامج , Constant Variable , Procedure , Program ويستخدم لتعريف الاقترانات المكتبية ، ومسمى المبرمج التعريفي (Identifier) يتكون من حروف وأرقام في أية ترتيب كان باستثناء الرمز الأول من المسمى والذي يجب أن يكون حرفاً مع إمكانية استخدام الحروف الصغيرة أو الكبيرة ومن المفضل استخدام الحروف الصغيرة كي يتم تمييزها عن الكلمات المحجوزة التي تستخدم في اكثر الأحيان الحروف الكبيرة ويشترط في بعض الأحيان بان لا يزيد طول الاسم التعريفي عن ثمانية خانات بينما

يسمح للبعض الآخر بأية طول كان ... وفي لغة Turbo Pascal يستخدم الرمز (-) في كتابة الاسم شريطة ان لايرد في بداية أو نهاية المسمى التعريفي للمبرمج.
وفيما يلي بعض الأمثلة للمسميات التعريقية الصحيحة والمقبولة بلغة Turbo Pascal :

a	grade	tax	b6	m7
average	f16	product	sum	length
flgure3	pressure	st-number	hours	rate
area	ss16			

أما الأسماء غير المقبولة في لغة Turbo Pascal فهي :

الاسم التعريفي	سبب عدم القبول
7-up	كونه يبدأ برقم وليس بحرف وكذلك لا يجوز استخدام الاشارة الخاصة (-)
3rd	كونه يبدأ برقم
B6.1	لكونه يستخدم الرمز الخاص (.)
X̄	لوجود رمز خاص فوق الحرف
VAR	لأنه كلمة محجوزة
B2	لاستخدامه حرف لاتيني

(4-2-9) الأسماء التعريفية القياسية Standard Identifiers

وفي بعض الأحيان تسمى المسميات التعريفية القياسية، وهي أسماء يشار إليها في كتابه البرامج ويكون لها معنى معرف مسبقاً ومنها الإقترانات القياسية المثلثية وكذلك الإقترانات الأسية والاقترانات الرياضية المكتبية والتي تسمى (Library Mathematical Functions) وكذلك الأسماء الدالة على نوع البيانات والاقترانات ويمكن إعادة تعريف هذه الأسماء القياسية على خلاف

الكلمات المحجوزة التي لا يمكن إعادة تعريفها ، وهذه الأسماء والاقترانات هي كما يلي :

ABS	IN	RESET
ARCTAN	MAXINT	REWRITELN
BOOLEAN	NEW	ROUND
CHAR	ODD	SIN
CHR	ORD	STRING
COS	OUTPUT	SQR
DISPOSE	PACK	SQRT
EOF	PAGE	SUCC
EOLN	PRED	TEXT
EXP	PUT	TRUE
FALSE	READ	TRUNC
GET	READLN	UNPACK
INPUT	REAL	WRITELN
INTEGER		

(9-2-5) الأعداد Numbers

الأعداد في Pascal هي عبارة عن سلسلة متصلة من الأرقام الحسابية والتي تبدأ بالعدد صفر 0 وتنتهي بالعدد 9 ويمكن للعدد أن يحمل الإشارة الموجبة أو السالبة ولا يمكن أن يحوي العدد على أية فارزة أو فراغ ، ويمكن للعدد أن يحوي الفاصلة العشرية (Decimal Point) أو أن تسبقه إشارة السالب (−) أو اشارة الموجب (+) والأعداد هي إما صحيحة أو حقيقية :

أ) **الأعداد الصحيحة Integer Numbers :**

العدد الصحيح في Pascal هو العدد الذي يكتب بدون فاصلة عشرية ويمكن أن يكون صفراً ، ويكتب دائماً بدون رمز الفاصلة العشرية .

والأمثلة التالية توضح الأعداد الصحيحة :

31   −20   −1000   +61   47   0

اما الأعداد غير المقبولة كأعداد صحيحة في لغة Pascal فهي على سبيل المثـــــال كما يلي :

العدد	سبب عدم القبول
31.3	احتوائه على الفاصلة العشرية
612,12	احتوائه على الفارزة
#50	احتوائه على الرمز الخاص #
C16	احتوائه على الحرف الابجدي c
221.	احتوائه على الفاصلة العشرية
251 25	احتوائه على فراغ

ب) الأعداد الحقيقية Real Numbers

الأعداد الحقيقية هي الأعداد الموجبة أو السالبة أو الصفر والتي تكتب بوجود الفاصلة العشرية ، ويمكن ان تكون هذه الأعداد في أحد الشكلين التاليين :

1- الشكل ذو الكسر العشري Decimal Notation

والأمثلة على العدد ذي الكسر العشري كما يلي :

-0.617   0.0   72.0   -14.07   50.16

ونشير هنا إلى أن العدد الحقيقي في الشكل ذي الكسر العشري يجـــب أن يحـــوي رمز الفاصلة العشرية ، وبالإضافة إلى أنه عند كتابة العدد الحقيقي في الشـــكل ذي الكسر العشري يجب وضع رقم عشري واحد على الأقل إلى يمين رمـــز الفاصلـــة العشرية وكذلك رقم عشري واحد على الأقل إلى يسار رمز الفاصلة العشرية .

و الأرقام التالية غير مقبولة بلغة باسكال كأعداد حقيقية وللأسباب الموضحة كمـا يلي :

العدد	سبب عدم القبول
4,561.1	لاحتوائه على الفارزة
6000	لايحتوي على الفاصلة العشرية
D7.2	يحتوي على حرف ابجدي
3 88.2	يحتوي على فراغ
.66	لا يحتوي على رقم إلى يسار الفاصلة
66.	لا يحتوي على رقم إلى يمين الفاصلة

2- الشكل المرفوع إلى أس Exponential Notation

وهي الأعداد المرفوعة إلى أس كما في الأمثلة التالية :

78.89E -36.8E6 16E5

ويستخدم هذا الشكل للأعداد المتناهية في الصغر أو المتناهية في الكبر فمثلاً يمكن تمثيل العدد الحقيقي التالي بالشكل الأسي وكما يلي :

0.000035 → 35.E-6
→ 35.0E-6
→ 3.5E-5
→ 0.35E-4

أو تمثيل العدد الحقيقي بالشكل التالي :

0.00234 → 2.34E-3

وهذا يعني ان العدد الموجود على يسار E مضروب بـ 10^{-3} ويمثل العدد الموجود إلى يمين E الأس الذي يرفع اليه العدد 10 ، مع ملاحظة أن الفاصلة العشرية المستخدمة في الشكل الآسي يجب أن يكون على الأقل رقم إلى يمينها ورقم إلى يسارها ، كما أن الأس المستخدم إلى يمين E يجب أن يكون عدداً صحيحاً .

أما مدى الثوابت العددية فتختلف من جهاز حاسوب إلى آخر حسب نوع وطراز جهاز الحاسوب المستخدم ، ، أما عدد الأرقام المعنوية المسموح بها في العدد الحقيقي فهي تختلف حسب رقم الإصدار للغة Pascal وفي معظمها يسمح

بـــ7 أو 8 أرقام . والجدول التالي يبين تمثيل بعض الأعداد بطريقة التدوين الأســـي بلغة باسكال :

التدوين الاساسي	التدوين الرياضي	القيمة
3.0E5	$3.0 * 10^5$	300000
3.0E05	$3.0 * 10^5$	300000
+3.0E5	$3.0 * 10^5$	300000
20.0E3	$20.0 * 10^3$	20000
7.63E-1	$7.63 * 10^{-1}$	0.763
6.0E-6	$6.0 * 10^{-6}$	0.000006

أما الامثلة التالية فهي غير مقبولة بالشكل الأسي وذلك للأسباب الموضحة لكـــل منها كما يلي :

العدد	سبب عدم القبول
21.5E3.4	لان الأس ليس عددا صحيحاً
7.E-4	لعدم وجود رقم إلى يمين الفاصلة العشرية
.6E-8	لعدم وجود رقم إلى يسار الفاصلة العشرية
8.8 E 3	لوجود فراغ في اس العدد إلى يمين E

(6-2-9) الثوابت الرمزية String Constants

الثابت الرمزي في لغة Pascal هو عبارة عن سلسلة من الرموز المكونة من الحروف والأرقام والرموز الخاصة باستثناء رمز الحاصرة العلوية (أو تسمى في بعض الأحيان علامتي الاقتباس) يسمى عدد الخانات المحجوزة بين الحـــاصرات العلويـــة بطول ايثابت الرمزي فمثلاً طول الثابت الرمزي "Ghazi Ibrahem" هـــو 13 رمزاً لأن الفراغ الموجود بين الاسمين يحسب رمزاً كونه احد رموز مجموعة الرمـــوز الخاصة ، والأمثلة التالية تمثل ثوابت رمزية صحيحة .

‘ The speed of mind ‘
‘ I love PASCAL ‘
‘ Applied science University ‘
‘ P. O. Box 40 ‘

كل الثوابت الرمزية أعلاه وإن استخدمت أرقاما حسابية داخلها (وذلك لأن هذه الأرقام لا تحمل أي قيمة أو معنى حسابي) تسمى بـالثوابت الرمزية ، وتستخدم عادة كعناوين توضيحية مع النتائج الخاصة بالبرنامج .

ويختلف الحد الأقصى لعدد الخانات المسموح بها للثابت الرمزي من نوع إصدار باسكال إلى اخر ويسمح معظم أشكال باسكال بطول مقداره 255 رمزا .

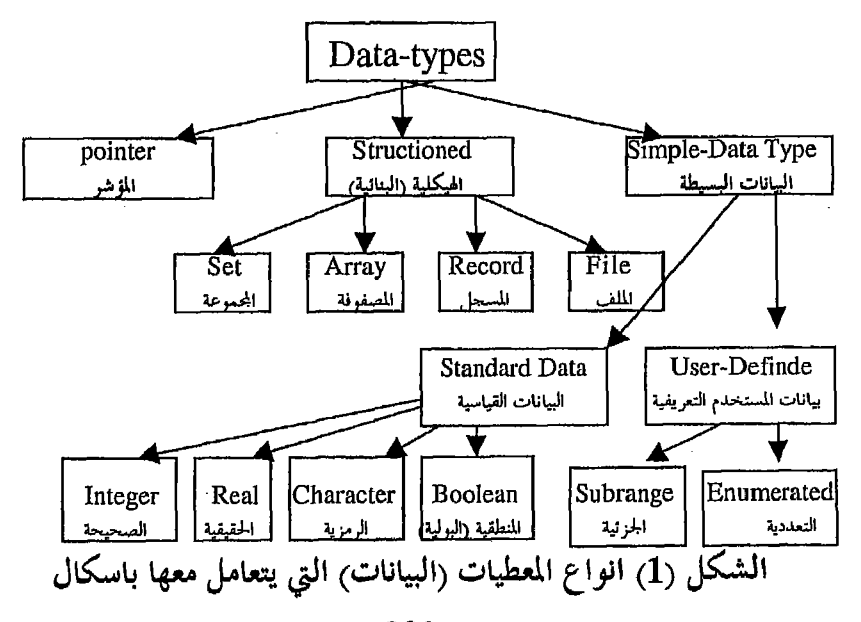
(9-2-7) أنواع المعطيات (البيانات) Data Types

تمتاز لغة Pascal عن لغات البرمجة الأخرى بقدرتها على التعامل مع أنواع مختلفة من المعطيات (البيانات) ، والشكل (1) يمثل انواع المعطيات .

وسنتعرض في هذا الفصل إلى المعطيات (البيانات) القياسية البسيطة والتي تعني بمتطلبات مقرر المقدمة ، ولمزيد من المعلومات عن الأنواع الأخرى يمكن العودة إلى المراجع المتخصصة بالبرمجة الهيكلية بلغة باسكال .

أ) الثوابت Constants

وهي التي تحمل قيماً ثابتة لا تتغير أثناء تنفيذ البرنامج سواء أكانت هذه القيم قيم حسابية أو رمزية وبحيث يتم تعريف قائمة الثوابت في مقدمة برنامج Pascal



الشكل (1) انواع المعطيات (البيانات) التي يتعامل معها باسكال

وذلك قبل استعمالها فيه ، ويمكن توضيح الصيغة العامة لجملة تعريف الثوابت بلغة Pascal كما يلي :

```
CONST name1 = value1 ;
      name2 = value2 ;
      name3 = value3 ;
              :
      namen = valuen ;
```

حيث ان الكلمة المحجوزة تم التعبير عنها بـــ : CONST

والاسم الثابت التعريفي بـــ : name

وقيمة الثابت الحسابية أو الرمزية بـــ : value

وفيما يلي بعض الأمثلة التوضيحية عن تمثيل الثوابت بلغة باسكال :

أ) الثابت الرمزي :

CONST address = ' science university

ب) الثابت الحسابي الحقيقي :

CONST distance = 16.4 ;

وسمي الثابت الحسابي حقيقي لان العدد في الطرف الأيمن في هذا المثال هو عدد حقيقي لذلك يعتبر الثابت في الطرف الأيسر distance هو حقيقي أيضاً :

ج) الثابت الحسابي الصحيح :

CONST length = 105 ;

وسمي الثابت الحسابي صحيح لان العدد في الطرف الأيمن في هذا المثال هو عدد صحيح ولذلك يعتبر الثابت في الطرف الأيسر Length صحيح ايضاً .

ومن أهم الثوابت العددية المحددة مسبقا في لغة باسكال هي كلمة MAXINT والتي تمثل أكبر عدد صحيح يستوعبه الحاسوب .

ب) المتغيرات Variables

المتغيرات هي التي تتغير قيمتها أثناء تنفيذ البرنامج ، والمتغير على عدة أنواع منها الحقيقي ومنها الصحيح ومنها الرمزي ومنها المنطقي ، لذلك يجب الاشارة إلى نوع المتغير في الجزء المخصص للتوضيح (أي الجزء الثاني في البرنامج الموضح في الفقرة 9-1) والصيغة العامة لجملة التعريف بنوع المتغير كما يلي :

```
VAR name1, name2, name3, .....namen : type;
      name1, name2, name3, ..... namen : type;
      ..............................
      ..............................
      name1, name2, name3,..... namen : type;
```

حيث : الاسماء التعريفية لاسماء المتغيرات (مسمى المبرمج التعريفي) هي :
name1 name2, name3, ...$name_n$

الكلمة المحجوزة لتعريف المتغيرات هي : VAR

ملاحظة :

type : تعني نوع المعطيات أو البيانات ويمكن ان تكون احدى الأنواع التالية :

متغيرات حقيقية REAL

متغيرات صحيحة INTEGER

متغيرات منطقية BOOLEAN

متغيرات رمزية CHAR

متغيرات سلاسل زمنية STRING

مثال (**1**) :

```
VAR
      Radius , Area : REAL ;
      Counter        : INTEGER ;
      Max            : CHAR ;
      State          : Boolean ;
```

حيث أن المتغيرات Radius , Area هما متغيرات حقيقيان ، و Counter هـــو متغير صحيح ، و Max هو متغير رمزي ، و State هو متغير منطقي .

(9-2-8) التعابير الحسابية والمنطقية Arithmetic and Expression

التعبير الحسابي هو مجموعة من الحدود المكونة من متغيرات وثوابت عدديـة صحيحة أوكسرية ، يربط بينها مجموعة من العمليات الحسابية .

أما التعبير المنطقي فهو مجموعة من الحدود المكونة من تعابير حســـابية أو ثوابت منطقية أو ثوابت رمزية ، يربط بينها معامل منطقـــي (مثـــل <، =>، =، <>،) ، والتعبير المنطقي قيمة إما أن تكون Yes أو أن تكون No والتعابير الحسابية يمكن أن تكون إما صحيحة أو حقيقية .

أ) التعابير الحسابية الصحيحة

يجب أن تكون جميع الحدود في التعبير الحسابي الصحيـــح مـــن النـــوع الصحيح والعمليات الحسابية يجب أن تكون نتيجتها أيضاً أعداداً صحيحة .

مثال (2) : بفرض أن A = 16 ، B = 12

إذا كانت العملية هي الجمع (+) فان التعبير A+B تكون نتيجته 28

إذا كانت العملية هي الطرح (–) فان التعبير A– B تكون نتيجته 4

إذا كانت العملية هي الضرب (*) فان التعبير B*A تكون نتيجته 192

إذا كانت العملية هي القسمة (Div) فان التعبير A/ B تكون نتيجته 1

إذا كانت العملية هي باقي القسمة (Mod) فان التعبير A mod B تكون نتيجته 4 وجميع هذه النتائج هي نتائج صحيحة .

مثال (3): أوجد ناتج العمليات الصحيحة التالية .

النتيجة	العملية
20	14+6
9	16-5
60	12*5
2	11 Div 5
1	11 Mod 5

مثال (4): أوجد ناتج العمليات الصحيحة التالية :

النتيجة	العملية
3	7 Div 2
1	4 Div 4
3	-7 2
-3	-7 Div 2
-3	7 2
7	17 Mod 10

مثال (5) : التعابير التالية غير مقبولة في لغة Pascal كتعابير صحيحة .

19 Div 14.0

7.5 Mod 2

ب) التعابير الحسابية الحقيقية :

يكون فيها حدي العملية الحسابية هو من النوع الحقيقي وتكون نتيجة العمليات الحسابية قيمة عددية حقيقية و العمليات الحسابية المستخدمة هي عمليات الجمع (+) ، والطرح (–) ، و الضرب (*) والقسمة (/)

مثال (6) : اذا كانت A = 4.5 ، B = 5.3 فإن :

A + B = 9.8

B -A = 0.8

B * A = 23.85

A / B = 0.85

ملاحظة : يمكن أن يكون أحد حدي العملية الحسابية صحيحاً والحد الآخر حقيقيا وفي هذه الحالة تكون نتيجة العملية الحسابية عدداً حقيقياً .

مثال (7) : إذا كانت A=11.2 ، B=5 فإن ناتج العمليات الحسابية الأربعة كما يلي :

A + B = 16.2

6.2

A * B = 56.0

A / B = 2.24

مثال (8) : التعابير التالية مقبوله في لغة Pascal على أساس أنها تعابير حسابية :

```
5*2 + 12/8
X*Y
X/Y
( a+b+c ) / ( a-b ) + c
( x-y ) + ( x+y ) / d
```

مثال (9) : التعابير التالية مقبولة في لغة Pascal على أساس أنها تعابير منطقية :

Y = X على اساس الجواب سيكون نعم أو كلا .

200.0 > Length على اساس الجواب سيكون نعم أو كلا .

Factor <= 1.0 على اساس الجواب سيكون نعم أو كلا .

ملاحظة :لا يمكن استخدام رمزين متتالين في تعبير واحد ، كذلك يجب معرفة أن لغة Pascal لا تستعمل الرفع إلى الاس بنفس الطريقة التي تستخدمها لغة

Basic أو Fortran وإنما تحول عملية الرفع إلى اس إلى عملية خطيــــة باستخدام اللوغاريتم.

كذلك تستعمل لغة Pascal الاقترانات القياسية (المكتبية)

Standard (Library) Functions

وهذه الاقترانات تكون جاهزة في الحاسوب للاستعمال الفوري ، وتســـمى في لغات البرمجة الأخرى الاقترانات المبنية داخلياً (   )

وعلى سبيل المثال يعتبر اقتران Cos(x) اقتران قياسي يمكن ايجاد قيمته حال إعطـاء قيمة (x) مع العلم انه يمكن أن تكون قيمة (x) بيان عددي أو منطقي أو رمزي أو حتى تعبير حسابي أو منطقي .

وتقسم الأقترانات القياسية إلى أربعة مجموعات :

1) اقترانات قياسية عددية نتيجتها أعداد صحيحة .

2) اقترانات قياسية عددية نتيجتها أعداد حقيقية .

3) اقترانات منطقية نتيجتها إما صح أو خطأ .

4) اقترانات رمزية تكون نتيجتها مستندة إلى النظام المستخدم في التشفير .

ولو عدّنا الآن إلى كل مجموعة من هذه الاقترانات لوجدنا ما يلي :

أولاً : ان الاقترانات القياسية العددية والتي تكون نتيجتها أعدادا صحيحة هي كما يلي :

- اقتران لحساب تربيع الأعداد الصحيحة SQR(x)
- اقتران لإيجاد القيمة المطلقة للعدد الصحيح ABS(x)
- اقترانات حذف الجزء الكسري للعدد الحقيقي TRUNC(x)
- اقترانات الجزء الصحيح في العدد الحقيقي INT(x)
- اقتران تقريب العدد الحقيقي إلى اقرب عدد صحيح ROUND(x)

مثال (10) :

ABS (5) = 5
ABS (-5) = 5
ROUND (4.5) = 4
ROUND (4.6) = 5
SQR (3) = 9
SQR (-3) = 9
TRUNC (4.2) = 4
TRUNC (4.6) = 4
INT (5.6) = 5

ثانياً : أما الاقترانات القياسية التي تكون نتيجة تنفيذها أعداداً حقيقية فهي جميـــع الاقترانات مثل :

- اقتران الجذر التربيعي SQR(x)
- اقتران القيمة المطلقة ABC(x)
- اقتران اللوغاريتم الطبيعي LN(x)
- اقتران تربيع العدد SQR(x)
- اقتران الظل المعكوس ARCTAN (x)
- الأقتران الأسي لحساب e^x EXP(X) .

مثال (11) :

ABS (3.2) = 3.2
ABS (-3.2) = 3.2
SQR (2.5) = 6.25
SQR (-2.5) = 6.25
SQRT (5) = 2.23
SIN (9) = 0.156
COS (-1.2) = 0.9998
LN (2) = 0.693
EXP (3) = 20.085
EXP (-3) = 0.0497

ثالثاً : أما الاقترانات المنطقية فهي التي يكون نتيجة تنفيذها إما True أو False وصيغتها العامة :

Gi R G2

حيث Gi : يمكن ان يكون تعبير حسابي أو ثابت منطقي أو رمزي .

R : معامل منطقي ويمكن أن يكون أصغر (<) أو أصغر أو يساوي (<=) أو يساوي (=) أو اكبر (>) أو اكبر أو يساوي(>=) أو لا يساوي (<>) كما يمكن ان يكون التعبير المنطقي مركباً وذلك عند استعمال كلمات NOT/AND/OR ومهما كان التعبير المنطقي بسيطاً أو مركباً فان قيمتها دائماً هي إما True أو False .

مثال (12) : إذا كانت قيمة a صح True

وقيمة b خطأ False

وقيمة c صح True

فإن قيمة التعبير المنطقي التالي صح True :

a AND (b OR c)

وقيمة التعبير المنطقي التالي خطأ False :

(a AND b) OR (b AND c)

وقيمة التعبير المنطقي التالي خطأ False :

a AND NOT b OR NOT c

رابعاً : أما الاقترانات الرمزية فهي الاقترانات التي تعتمد على وضع حاصرتين علويتين ولهما علاقة بنظم الشيفرة لتبادل المعلومات مثل ASCII المستخدمة مع معظم اجهزة الحواسيب الشخصية (أنظر الجدول (1) الفصل الرابع) وهذه الاقترانات هي :

- اقتران PRED(x) وهو اقتران تحديد الرمز الذي يسبق الرمز (x) في مجموعة الرموز ويسمى Predecessor function

- اقتران SUCC(x) وهو اقتران تحديد الرمز الذي يلي الرمـــز (x) في مجموعة الرموز ويسمى Successor Function .

- اقتران ORD(x) وهو اقتران ايجاد القيمة الداخلية للرمز (x) ويسمى Ordinary Function .

- اقتران CHR(x) وهو اقتران إيجاد الرمز الخارجي الذي يقابل العـدد الصحيح (x) ويسمى Character Function

(9-2-9) قاعدة الأولوية Rule of Precedence

قاعدة الأولوية هي القاعدة المستخدمة في معالجة العمليات الحسابية وفـــق تسلسل خاص بحيث يتم معالجة البيانات وبالتسلسل التالي :

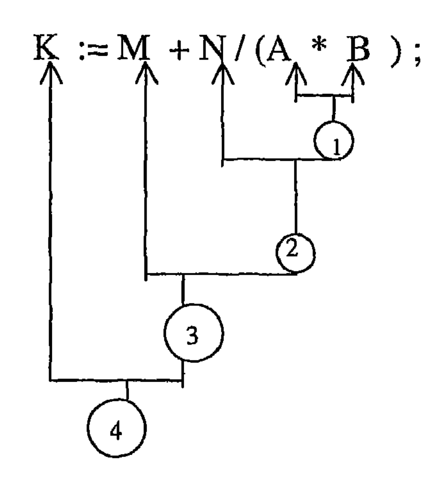
1. معالجة ما داخل الأقواس .
2. معالجة عملية الضرب (*) وعملية القسمة (DIV) وعملية الباقي من القســـمة (MOD) وعملية AND وعملية (/) وهذه العمليات لها نفس الأولوية .
3. معالجة عملية الجمع (+) والطرح (–) وعملية OR ، ولهما نفس الأولوية .

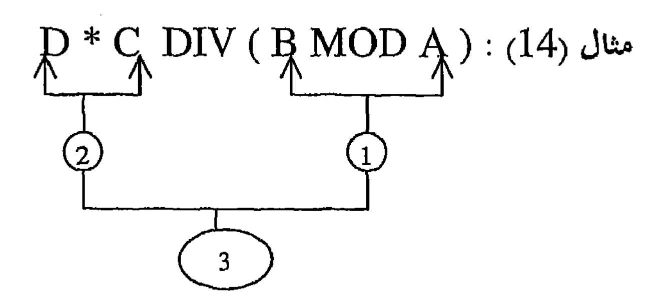
مع الأخذ بعين الإعتبار ما يلي :

أ– العمليات الحسابية التي لها نفس الأولوية تنفذ على الترتيب حسب ظهورها مـــن اليسار إلى اليمين .

ب – اذا كانت الأقواس متداخلة تعطي الأولوية في التنفيذ إلى القوس الداخلـــي ثم الخارجي الذي يليه ثم الخارجي الذي يليه، أي أن نبدأ من الداخل الى الخارج .

مثال (13) : لاستخدام قاعدة الأولوية في معالجة المعادلة التالية تصبح كالاتي :





(9-2-10) جمل الإسناد Assignment Statement

- جملة الإسناد (جملة التعيين) تستعمل لتخصيص قيمة محددة إلى متغير ، والصيغة العامة لجملة الاسناد في لغة باسكال هي :

Variable := Expression

حيث ان :

Variable: هو متغير قد يكون من أي نوع .

Expression : هو تعبير وقد يكون ثانيا (عدديا أو منطقيا) أو متغـــيرا من نفس نوع المتغير الموجود في الطرف الأيسر .

مثال (15) : الجملة التالية هي جملة إسناد حسابية

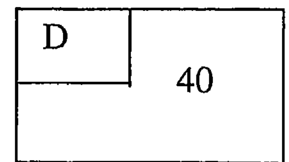
Zakat := 0.025 * Total Money ;

حيث الفارزة ; المنقوطة تعني فصل هذا السطر عن السطر اللاحق فلو فرضنـا ان Total Money كانت قيمته مساوية إلى 1000 مثلا فإن النتيجة 25 سوف يتم خزنها في مخزن المتغير الخاص بـ Zakat

مثال (16) : جملة الاسناد التالية :

D := 40 ;

تعني اسناد القيمة العددية 40 للمتغير العددي D



مثال (17) :جملة الإسناد الحسابية التالية :

Number := Number + 1 ;

تعني إضافة القيمة 1 إلى قيمة المتغير Number وإسناد النتيجـة ثانيـة للمتغـير Number وهذا يعني أن جملة الإسناد تصبح فيها قيمة المتغير Number الجديـدة تساوي قيمة Number مضافا إليها 1 .

(9-3) المكونات الهيكلية لبرنامج باسكال:

Structure Of Pascal Program

كما أوضحنا سابقا فان أي برنامج بلغة باسكال يتكون من ثلاث مقاطع رئيسيه :

المقطع الأول : وهذا المقطع مخصص لعنوان البرنامج أو ما يسمى :

The Program Heading

المقطع هذا يتكون في كثير من الأحيان من سطر واحد على شرط أن يبدأ بكلمـة

محجوزة وهي كلمة PROGRAM ثم يليها اسم البرنـامج (program-name) مع ملاحظة ان هذا المقطع يجب أن يخضع في كتابته إلى قواعد كتابـــة المســميات التعريفية ، ثم يتبع اسم البرنامج وعنوانه اسلوب أو نوع ملف الإدخال والإخـــراج ويمكن في بعض الحالات عدم تحديد اسلوب ملـــف الإدخـــال في حالـــة كـــون الاسلوب معروف من قبل الحاسوب والصيغة العامة للمقطع الأول كما يلي:

ROGRAM x-name (INPUT,OUTPUT,FILE-NAME,..)

وفي بعض الأحيان يمكن اضافة سطر توثيقي محصور بـــين حـــاصرتين وقوســـين للتعريف باسم ونوع وعمل البرنامج وهذا السطر يكون فقط للشرح .

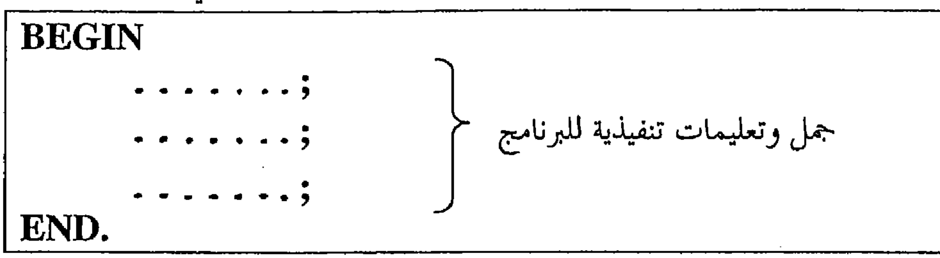
ونشير هنا إلى أننا نكتب كلمة الادخال INPUT في عنوان البرنامج في حالـــة كون تنفيذ البرنامج يتطلب قراءة بيانات من لوح المفاتيح (Keybord) وكلمـــة (OUTPUT) في حالة كون تنفيذ البرنامج يتطلب إظهار النتائج علـــى وســـط الإخراج من خلال جهاز الإخراج ، علما يأن هذين الملفـــين ملـــف الإدخـــال INPUT وملف الإخراج OUTPUT هما قياسيان ولا يجوز تعريفهم تـــأتي في بداية السطر . أما فهو يمثل اسم الملف الخارجي والذي يتـــم تعريفه ضمن البرنامج مع ملاحظة أنه في حالة وجود اكثر من ملف واكثر من اسم واحد للملف فيجب كتابه جميع اسماء الملفات و يجب أن نضع الفاصلة (,) بين كل اسم ملف وآخر .

<u>المقطع الثاني</u> : يسمى هذا المقطع بمقطع التوضيحات أو التعريفات أو التصريحـــات وهي (Declartion Section) ويتكون هذا المقطع من جملة واحدة أو أكثر مـــن جملة وذلك للتعريف أو للاعلان عن العناوين ، أو البيانـــــات ، أو الثوابـــت ، أو الإجراءات، أو الاقترانات المستخدمة في البرنامج مع ضرورة معرفة التتابع المنطقـــي للتعريفات حسب التسلسل التالي :

LABEL
CONST
TYPE
VAR
PROCEDURE
FUNCTION

حيث أن هذا التعريفات يجب أن تكون متسلسلة كما في أعلاه في حالـــة احتـــواء البرنامج الواحد على أكثر من تصريح أو توضيح .

<u>**المقطع الثالث**</u> : وهو المقطع التنفيذي (Executable Section) أو ما يســمى في بعض الأحيان Bluk ، ويضم تعليمات وجمل البرنامج المطلوب تنفيذها لتحقيق الهدف المراد من تصميم البرنامج وتنفيذه ، ويكون هذا المقطع دائما محصور بـــين الكلمتين المحجوزتين BEGIN و END ، والصيغة العامة كما يلي:



حيث أن كلمة BEGIN تمثل بداية البرنامج ولا يتبعها أية اشارة وتمثـــل بدايـــة البرنامج التنفيذي بينما ينتهي البرنامج بكلمة END يتبعها نقطة (.) إلزامية وهـــي تمثل نهاية البرنامج .

أما الجمل والتعليمات التي تأتي داخل الكلمتين BEGIN و END ، فيجـــب ان تنتهي بفاصلة منقوطة (;) Separator ، وهنا نشير إلى ان القسم التنفيذي يمكـن ان يضم اكثر من BEGIN و END كمقاطع أو أجزاء داخلية ، وبالتالي يجـــب ان تنتهي كل END بفاصلة منقوطة ما عدا END الاخيرة التي تمثل نهاية البرنامج فتنتهي بـــ نقطة .

مثال (18) : المثال التالي يوضح برنامج بلغة باسكال لايجاد مساحة ومحيط دائـــرة نصف قطرها معلوما .

الخطوة الأولى : قانون مساحة الدائرة هو :

المساحة = نصف القطر * نصف القطر * النسبة الثابتة

Area := Pi * SQR(Radius)

باعتبار ان Pi = 3.14159 وهي النسبة الثابتة

باعتبار ان (Radius) هو نصف القطر

قانون محيط الدائرة هو :

المحيط = النسبة الثابتة * نصف القطر * 2

Circumference := 2* Pi * Radius ;

باعتبار ان Circumference هو محيط الدائرة .

اذا هذه المسألة احتوت على ثوابـــت وهـــي النســـبة الثابتـــة : Pi=3.14159 والمتغيرات هي نصف قطر الدائرة المعلوم Radius كمدخلات ، ومساحة الدائـــرة Area ومحيط الدائرة Circumference كمخرجات ويمكن ان يكون نصـــف قطر الدائرة كمخرجات بالإضافة إلى كونه كمدخلات ، والعلاقات هي القوانـــين الخاصة بالمساحة والمحيط للدائرة .

الخطوة الثانية : تحديد ما تم شرحه في الخطوة الأولى وبالشكل التالي :

الثوابت : Pi=3.14154

المتغيرات :

المدخلات: Radius : نصف القطر

المخرجات: Area: مساحة الدائرة

Circumference : محيط الدائرة

Radius : نصف القطر

إذاً البرنامج يكون على الشكل التالي حسب المقاطع الثلاث التي ذكرناها سابقا والتي يتكون منها برنامج Pascal :

المقطع الأول

```
PROGRAM GHAZI ( Input , Output ) ;
(*PROGRAM to compute and print the Area and
    Circumference of a given circle*)
```

سطر تعريف غير ملزم وغير منفذ

المقطع الثاني التعريفات أو التصريحات

```
CONST
     Pi = 3.14 159 ;
VAR
          Radius ,AREA ,Circumference : REAL ;
```

المقطع الثالث المقطع التنفيذي

```
BEGIN
          WRITELN ('ENTER The Value of Radius') ;
          READ (Radous) ;
          AREA := Pi * SQR(Radius) ;
          Circumference := z* Pi * Radius ;
          WRITELN('FOR Radius = ', Radius ) ;
          WRITELN('The Area    = ' , Area ) ;
          WRITELN('Circumference=',circumferenc);
END .
```

ثم بعد كتابة وادخال البرنامج إلى الحاسوب يتم اعطاء قيمة نصف القطر Radius ثم يتم الضغط على مفتاح الإدخال ENTER وتظهر النتائج على الشاشة .

(9 – 4) أساليب إدخال وإخراج البيانات في برنامج باسكال

ستتعرض في هذه الفقرة إلى الأساليب المتبعة في نقل وإدخال البيانات ذات القيم الحسابية أو الرمزية وكذلك سوف نتعرض إلى أساليب إخـــراج وطباعـــة نتائج المدخلات بعد إجراء المعالجات عليها .

(9–4–1) جمل الإدخال Input Statements

يتم إدخال البيانات إما عن طريق ادخال قيم البيانات مع البرنامج نفسه باستخدام جملة الاسناد Assignment Statement وتسمى أيضا جملة التعيين ، أو عن طريق تسجيل البيانات في ملف يدعى ملف المدخلات حيث يتم قراءة قيم المتغيرات منه باستعمال جمل القراءة أثناء تنفيذ البرنامج أو بالإدخال المباشر باستخدام جمل القراءة أيضا .

أولا : استخدام جملة الإسناد

كما وضحنا في الفقرة (9–2–10) فإن جملة الإسناد تأخذ الشكل التالي :

VARIBLE := EXPRESSION

وهذا يعني أنه يتم إسناد قيمة التعبير Expression الموجود في الطرف الأيمن إلى المتغير الموجود في الطرف الأيسر ، وتخزن قيمة التعبير باسم المتغير ، ويتم التعامل في البرنامج أينما وجد المتغير مع قيمته المخزنة . وعادة يستخدم هذا الأسلوب عندما تكون عدد المتغيرات قليلا والبرنامح صغيرا .

مثال (19) :

Sum := 5032.5

حيث يتم تخزين العدد 5032.5 باسم المتغير Sum ، وبالتالي أينما وجدت كلمة Sum في البرنامج يتم تبديلها بقيمتها مباشرة . المؤشرة أعلاه

مثال (20) : إذا كان :

S := 50;
K := 5;

عندئذ عند ظهور الجملة التالية في البرنامج .

mean := S/N;

يتم تبديل S بـــ 50 و N بـــ 5 وتجري عملية القسمة ويخزن الناتج الذي يساوي 10 في المتغير mean .

ثانيا : استخدام ملف المدخلات :

يستخدم هذا الأسلوب عندما تكون عدد المتغيرات في البرنـــامج كثـــيرة وحجم البيانات كبيرا ، حيث تقرأ البيانات الموجودة على ملف الإدخـــال Input File بصورة آلية من خلال الجمل والأوامر الخاصة بقراءة الملفات ، وأهم مـيزات هذا الأسلوب هو إمكانية تنفيذ البرنامج عددا مـــن المـــرات يســـاوي إلى عـــدد السجلات الموجودة في الملف حيث يقوم البرنامج بتكرار الخوارزمية المطلوبة مـــن أجل جمع بيانات الملف .

ثالثا : التخاطب المباشر مع الحاسوب :

وذلك باستخدام جملة القراءة READ , READLN

أ. جملة القراءة (الإدخال) **READ Statement** :

هذه الجملة تستخدم في لغة Pascal لقراءة وإدخـــال قيـــم المتغـــيرات الحسابية أو الرمزية إلى الحاسوب لاستخدامها في البرنامج وصيغتها العامة كمايلي:

$\text{READ (Variable}_1\text{, Variable}_2 \quad \text{Variable}_n\text{)}$

حيث نجد أن جملة READ تمثل مسمى تعريفي قياسي ويعامل معاملـــة أحد الكلمات المحجوزة أما (Variable) فهي إحدى مسميات المـــبرمج لاسمـــاء المتغيرات وقد تكون متغيرات عددية أو رمزية .

كما نلاحظ في الصيغة العامة اعلاه ان اسم المتغير المراد ادخال قيمته الحســـابية أو الرمزية تكون موضوعة ومحصورة بين قوسين وتسبقها جملة READ مع ملاحظـة هامة ، وهي اننا قمنا باستثناء المتغير المنطقي وذلك لعدم امكانية ادخـــال المتغـــير المنطقي باستعمال جملة القراءة READ .

ويتم تنفيذ أمر قراءة المتغيرات بالترتيب الذي وضعت عليه وحســـب الوصـــف المعرفة فيه وبشكل تسلسلي ومن خلال إدخال هذه القيم من لوحة المفـــاتيح مـــع مراعات بعض الملاحظات التي سنوضحها من خلال الأمثلة التالية :

مثال (21) :ليكن لدينا البرنامج التالي الذي يضم بعض جمل القراءة .

```
PROGRAM Exam (Input,Output);
VAR
      F,C,N,S,L: INTEGER;
BEGIN
      READ (N,C,F);
      S:=F + C;
      L:=N +F;
      WRITELN ('S=',S,'L=',L);
END.
```

ولدى تنفيذ البرنامج تظهر شاشة المخرجات (الشاشة السوداء) ويظهر المؤشر (–) coursor في الجهة اليسرى من الشاشة ، ويجب على المستخدم إدخال قيم المتغيرات من لوحة المفاتيح بنفس التسلسل التي وضعت فيه في جملة القراءة إما على سطر واحد على أن يفصل بينها مسافة لا تقل عن فراغ واحد ثم ضغط مفتاح الإدخال Enter ليتم تخزين تلك القيم المدخلة باسم المتغيرات ثم متابعة تنفيذ البرنامج ، أو إدخال كل قيمة على سطر يتبعها ضغط مفتاح الإدخال Enter .

أي ان شاشة الإخراج عند تنفيذ البرنامج ستكون على الشكل التالي :

```
4   5   6 ↵
S=11     L=10
```

أو

```
4 ↵
5 ↵
6 ↵
S=11     L=10
```

حيث اشارة ↵ تعني أنه يجب ضغط مفتاح الإدخال Enter .

ملاحظة : إن المتغيرات في جملة القراءة في المثال (21) السابق يمكن كتابتها ضمن اكثر من جملة قراءة وعلى الشكل التالي :

```
READ (N,C);
READ (F);
```

أو

```
READ (N);
READ (C,F);
```

أو

```
READ (N);
READ (C);
READ (F);
```

وعند التنفيذ يتم إدخال قيم المتغيرات كما ذكرنا سابقا إما على سطر واحد يفصل بينها فراغ واحد على الأقل ثم مفتاح ENTER أو كل قيمة على سطر يتبعها ENTER .

ب. جملة القراءة (الإدخال) READLN Statement

لا تختلف هذه الجملة عن جملة الإدخال READ في وظيفتها وإنما تختلف من حيث الأداء ، حيث أن READLN تعني القراءة على نفس السطر الموجود عليه المؤشر والصيغة العامة لها نفس الصيغة للجملة READ .

مثال (22) : إذا كتبنا جملة القراءة في البرنامج السابق المثال (21) بالشكل التالي:

```
READLN (N,C,F);
```

فإنه يتم الإدخال بنفس الاسلوب الموضح في المثال (21) ، اما إذا كتبنا جملة القراءة بالشكل التالي :

```
READLN(N,C);
READ (F);
```

عندئذ يجب إدخال قيمة N,C على السطر الأول بحيث يفصل بينها فراغ ثم الضغط على مفتاح ENTER ثم قيمة F على السطر التالي ، وإذا حدث وأن كتبنا القيم الثلاث على سطر واحد فإنه لا يأخذ حسب الجملة الأولى إلا قيمتين فقط ويطلب من المستخدم إدخال القيمة الثالثة على سطر جديد.

هذا يعني إن وجود LN في نهاية الجملة READ تعلم الحاسوب بنقل ميكانيكية قبول المدخلات من السطر التالي الذي يليه .

ملاحظة : عند كتابة جملة الإدخال بالشكل التالي :

READLN;

في نهاية البرنامج وقبل جملة END الأخيرة ، تعطي إمكانية رؤية نتيجة تنفيذ البرنامج على شاشة المخرجات ، على أن يسبقها جملة READLN في البرنامج بأي شكل ، وبالضغط على مفتاح Enter نعود إلى شاشة باسكال المكتوب فيها البرنامج .

(2-4-9) جمل الإخراج OUTPUT Statement

يتم إخراج نتائج تنفيذ البرنامج إما على الشاشة مباشرة أو على الطابعة أو تخزين النتائج على ملف يدعى ملف الإخراج Output Flie، وفي جميع الحالات نستخدم الجملة WRITE أو WRITELN مع اختلافات بسيطة حسب نوع الإخراج .

أ. جملة الإخراج غير الموصوفة WRITE Statement

تستخدم هذه الجملة لإخراج النتائج على الشاشة والصيغة العامة لها كما يلي :

```
WRITE (List) ;
```

حيث أن list تمثل ثوابت أو متغيرات Variables أو تعابير حسابية أو منطقية Expression ، كما يمكن أن تحتوي على جمل و تعليقات لشرح نوعية المخرجات على أن توضح هذه التعليقات ضمن حاصرتين علويتين وليس بالضرورة أن تضم list نوع واحد كما ذكر آنفا بل يمكن أن تكون خليط من جميع الأنواع وهذا يتعلق بالمبرمج وشكل النتائج المراد اخراجها ، ويفضل دوما كتابة

تعليقات قبل كل نوع ثمن ماهية النتائج وخاصة عند التعامل مع برامج تضم حجم كبير من النتائج .

ونشير هنا إلى أن قائمة list من الأنواع السابقة يجب أن تفصل بينـــها الفـــارزة (commans) .

مثال (23) :اذا كانت قيم المتغيرات التالية في البرنامج كما يلي :

A = 10 , B = 20 , C = 30 , D = 40 , E =50 , عندئذ باستخدام الجملة التالية في البرنامج

```
WRITE (A,B,C,D,E) ;
```

تظهر النتائج على شاشة المخرجات كمايلي على سطر واحد

```
1020304050
```

نلاحظ أن النتائج لا تفصل بينها أية فراغات ، لذلك يجب وضع فراغات بينها عن طريق ترك فراغ او أكثر ضمن حاصرتين ضمن جملة الطباعة بالشكل التالي :

```
WRITE (A,' ',B,' ',C,' ',D,' ',E);
```

```
10 20 30 40 50
```

مثال (24) لطباعة قيم تعبير حسابي باستخدام جملة الطباعة WRITE بفرض أن قيمة a=7 , b=5 نستخدم جملة الطباعة التالية :

```
WRITE ('D=' , a-b) ;
```

فإن شكل الطباعة كنتيجة سوف تظهر كما يلي :

```
d = 2
```

مثال (25) : لطباعة قيم لثوابت رمزية باستخدام جملة الطباعة WRITE على الشكل التالي :

```
WRITE ('GHAZI',' MOUHAMAD',' IBRAHEM')
```

فإن النتيجة ستكون بالشكل التالي :

```
GHAZI   MOUHAMAD   IBRAHEM
```

مع ملاحظة الفراغات فكلما زادت الفراغات داخل الحواصر العلوية لكل ثابت رمزي كلما زادت الفراغات في الطباعة واظهار .

مثال (26): في بعض الأحيان يطلب اظهار تعليقات على المتغيرات مع قيمتها عند الطباعة فلو فرضنا ان قيم A=15.2 , B=62.2 , C=14 , D=20 , E=G

وكانت جملة الطباعة WRITE بالشكل التالية :

```
WRITE ('A=',A,'  B=',B,'  C=',C);
WRITE ('  D=,D,'  E=',E);
```

عندئذ تظهر النتائج على شاشة المخرجات كما يلي :

```
A =15.2  B =62.2  C =14  D =20  E =6
```

ب) جملة الإخراج غير الموصوفة WRITELN Statement

إن جملة الطباعةWRITELN والتي تعني الطباعة بسطر جديد لا تختلف كثيرا عن جملة الطباعة غير الموصوفة WRITE في وظيفتها اذ ان الاختلاف هو في طريقة الأداء في الطباعة حيث ان الصيغة العامة لهذه الجملة هي كما يلي :

```
WRITELN (List) ;
```

حيث List تمثل قوائم الطباعة والإخراج والتي من الممكن أن تكون واحد او اكثر من الأنواع التالية :

1. Variables	متغيرات حقيقية أو صحيحة أو رمزية
2. Constatants	ثوابت
3. Strings	ثوابت رمزية
4. Expressions	تعابير حسابية أو منطقية

على أن يفصل بين هذه المكونات الفارزة (,) (Commons) .

إن وجود LN في نهاية الجملة WRITE يعطي الامر إلى الحاسوب بنقل ميكانيكية الطباعة إلى السطر التالي .

ملاحظة (1) : ان استخدم امر الطباعة السابقة بالشكل التالي :

```
WRITELN;
```

يؤدي إلى ترك سطر فارغ ونقل ميكانيكية الطباعة إلى السطر التالي .

مثال (27) : اذا كانت قيم :

. A= 4, B= 6, C= 12.5, D= 12, E= 14, F= 5.2

فعند استخدام جملة WRITELN التالية :

```
WRITELN (A,B,C,D,E,F) ;
```

فإن النتيجة ستظهر كما يلي :

```
4612.512145.2
```

وعند استخدام جملة WRITELN بالشكل التالي :

```
WRITELN (A) ;
WRITELN (B);
WRITELN (C,D);
WRITELN (E,'  ',F) ;
```

فإن النتيجة ستظهر كما يلي :

```
4
6
12.512
14   5.2
```

ج) جمل الطباعة الموصوفة Formatted Writeln

لتوصيف البيانات المخرجة والتحكم بشكل الإخراج نستخدم جملتي WRITE ، WRITELN بنفس الاسلوب السابق مع بعض الإضافات حسب نوع البيانات .

لطباعة العناوين والثوابت الرمزية ، نستخدم الصيغة التالية :

```
WRITE ('String Constant':N,........ ) ;
```

حيث : String Constant ثوابت رمزية .

N طول الحقل المخصص للطباعة .

ويمكن استخدام WRITELN أيضاً بنفس الصيغة

مثال (28) :

```
WRITELN ('1245 AY');
WRITELN ('PASCAL':10,' ':5,'Programming':10);
```

وعند التنفيذ تكون شكل المخرجات كما يلي :

1	2	4	5	A	Y																			
				p	a	s	c	a	l					p	r	o	g	r	a	m	m	i	n	g

من خلال المثال السابق نلاحظ ان الجملة الاولى هي بدون توصيف وظهرت النتائج ابتداءً من العمود الأول ، بينما في الجملة الثانية عندما وصفنا الثابت الرمزي الاول ب 10 عندئذ يقوم الحاسوب بحجز العشرة أعمدة الأولى لوضع كلمة Pascal ويكتبها في أقصى اليمين من الحقل المخصص ، ثم يحجز خمسة أعمدة ويتركها فارغة ، ثم يقوم بحجز عشرة أعمدة أخرى لكلمة Programming ، إلا ان الكلمة الأخيرة تحتاج الى اكثر من عشرة اعمدة عندئذ يهمل الحاسوب طول الحقل المخصص ويكتبها ضمن (11) عامود .

لطباعة الأعداد الصحيحة ، نستخدم الصيغة التالية :

```
WRITE (Variable:N,....... ) ;
```

حيث Variable يمثل اسم المتغير المعرف على انه صحيح Integer

N هو طول الحقل المخصص لطباعة العدد الصحيح بما فيه الاشارة

مثال (29) : اذا كان :

A=8451,   B=-89,   C=12354

فإن أمر الطباعة التالي :

```
WRITELN (A:6,B:4,' ' :3,C:3);
```

يظهر عند التنفيذ النتائج بالشكل التالي :

		8	4	5	1		-	8	9				1	2	3	5	4			...

من هذا المثال نلاحظ ان الحاسوب حجز 6 أعمدة الأولى لوضع قيمةA ويكتبها في أقصى اليمين من الحقل المخصص لـ A ، ثم يحجز الاعمدة الثلاثة التالية ويتركها فارغة ، ثم يحجز الاعمدة الثلاثة التالية لوضع قيمة C والتي تحتاج الى خمسة أعمدة عندئذ الحاسوب يتجاهل التوصيف المحدد بجملة الطباعة ويكتب قيمة C ابتداءً من العمود الذي يلي الحقل المخصص للفراغات الثلاث .

ولطباعة الأعداد الحقيقية ، تستخدم الصيغة التالية :

```
WRITE (Variable: N:M,....);
```

حيث : Variable هو اسم المتغير المعرف على أنه حقيقي Real .

N طول الحقل المخصص لطباعة العدد الحقيقي بجزئية الصحيح والكسري والفاصلة العشريه والإشارة .

M طول الحقل المخصص لطباعة الجزء الكسري فقط .

مثال (30) : اذا كان :

X=28.738, Y=132.4,   Z=5.6

فإن أمر الطباعة التالي :

```
WRITELN (X:6:2,'    ', Y:4:2,' ':3,Z:4:2);
```

	2	8	.	7	4					1	2	3	.	4	0				5	.	6	0		..

من المثال نلاحظ ان الحاسوب من خلال التوصيف سيقوم بحجز الأعمدة الستة الأولى لوضع قيمة X ، ويكتبها في أقصى يمين الحقل المخصص .

مع الأخذ بعين الاعتبار ان القسم الكسري مخصص له فقط خانتين أي عمودين لذلك يقوم الحاسوب تلقائيا بتدوير الجزء الكسري الى خانتين فقط ، يلي الاعمدة الستة الأولى أربعة أعمدة فارغة ثم يليها اربعة اعمدة لقيمة Y ، الا ان قيمة Y تحتاج الى اكثر من اربعة اعمدة لذلك يتجاهل الحاسوب التوصيف ، ويقوم فقط بتوصيف الجزء الكسري أي انه يكتب الجزء الصحيح بأكمله وفي الجزء الكسري خانتين بمعنى آخر ان قيمة Y احتاجت الى 6 أعمدة ، بعد قيمة Y مباشرة بترك ثلاثة أعمدة فارغة ثم يحجز بعدها أربعة أعمدة لوضع قيمة Z حيث يكون مخصص فقط عمودين للجزء الكسري .

ملاحظات حول قواعد التوصيف :

1- يتم طباعة البيانات في أقصى يمين الحقل المخصص في التوصيف .

2- اذا كان عدد رموز الثابت الرمزي او عدد خانات العدد المراد طباعتها اكبر من عدد الاعمدة المخصصة له ، عندئذ يتجاهل الحاسوب التوصيف ويطبعها وكانه لا يوجد توصيف أصلا .

3- إذا كان عدد خانات الجزء الكسري في العدد الحقيقي المراد طباعته اكبر من عدد الاعمدة المحجوزة له في التوصيف عندئذ يقوم الحاسوب بتدوير الجزء الكسري حسب الطلب .

4- اذا كان عدد خانات الجزء الكسري في العمود الحقيقي المراد طباعته اقل من عدد الاعمدة المحجوزة له في التوصيف عندئذ تملئ الأعمدة الزائدة بأصفار .

د) فتح صفحة جديدة Page

باستخدام جملة الاخراج او الطباعة WRITELN وفي حالة الحاجة لطباعة الجملة في صفحة جديدة يمكن ان نستخدم الجملة التالية للطباعة :

```
WRITELN (C,D,E);
PAGE;
WRITELN (E,F,G);
```

في هذا المثال نلاحظ انه سيتم طباعة C,D,E في سطر فيما يتم طباعة C,D,E في سطر جديد وصفحة جديدة ، علماً أن إصدارات باسكال ليست جميعها تقبل صيغة الصفحة الجديدة هذه .

مثال (31): افترض أن قيم المتغيرات هي كما موضح صورها أدناه وضح كيف ستكون صورة طباعتها باستخدام جملة الطباعة الموضحة .

المتغير	نوع المتغير	قيمة المتغير
E	صحيح	-80
F	حقيقي	11.2
G	رمزي	GHAZI
H	منطقي	FALSE

باستخدام جمل الطباعة التالية :

```
WRITELN ('E=' , E:4,' ':3);
WRITELN ('F=' , F);
WRITE ('G=' , G);
WRITELN ('H=' , H);
```

ستكون النتيجة كما يلي :

E	=		-	8	0				F	=	1	1	.	2		
G	=	G	H	A	Z	I	H	=	F	A	L	S	E			..

مثال (32) : افترض ان برنامج باسكال مكتوب بالشكل التالي ..نفذ هذا الشكل واظهر النتائج (الإخراج) .

```
PROGRAM GHAZI (Output) ;
VAR A , B : INTEGER ;
      C , D : REAL ;
BEGIN
      A:=4 ;
      B:=6 ;
      C:=10.2 ;
      D:=25.4 ;
WRITELN (A,B:3,A+B:4) ;
WRITELN (A+9) ;
WRITELN (D);
END.
```

النتائج سوف تكون كما يلي :

4			6			1	0						
1	3		1		0	2	0	0	E	+	0	1	
2	5		4										

(5-9) كتابة وتصميم برنامج Pascal

من الممكن كتابة برنامج بسيط بلغة باسكال بعد أن وضحنا جمل الادخال (القراءة) وجمل الإخراج (الطباعة) وصولا إلى ادخال جمل جديدة في برنامج Pascal ، وقبل كتابة برامج بلغة Pascal لابد من معرفة الهدف النهائي المطلوب تحقيقه ، حيث نبدأ بعملية شرح وفهم البرنامج المطلوب كتابته من خلال كتابة مجموعة من الخطوات المنطقية الرئيسية المترابطة فيما بينها وهذا ما يعرف بالخوارزمية كما أشرنا في الفصل الثامن ونستخدم احيانا اخرى ما يسمى بالمخطط الإنسيابي(Flowchart) والذي يوضح انسيابية البرنامج المهيكل ، ثم بعد كتابة وفهم هذا المخطط أو الخوارزمية نقوم بنقل هذا المخطط من صورته الحالية إلى

صورة برمجية طبقا للايعازات والخطوات المعروفة في جمل لغـــة Pascal ولنـــأخذ المثال التالي لتنفيذ ما سبق .

فلو فرضنا إننا نروم كتابة برنامج بلغة Pascal لإيجاد مساحة دائرة نصف قطرهـــا معلوم ، فإن الخطوات المتسلسلة المنطقية والتي تسمى الخوارزمية التي نستطيع مـــن خلالها الوصول إلى ايجاد مساحة الدائرة هي كما يلي :

الخطوة الأولى : ما هي المدخلات .

الخطوة الثانية : ما هي المخرجات .

فلو تفحصنا جيدا مثالنا اعلاه لوجدنا ان الخطوة الأولى سوف تشتمل على ما يلي:

1. نصف القطر والذي يمثل : Radius
2. النسبة الثابتة والتي تمثلها : Pi والتي تساوي 3.14154
3. استخدام قانون المساحة والتي يمثلها القانون Pi*SQR(Radius)

اما الخطوة اعلاه فتمثل الثانية تمثل في مثالنا هذا المساحة والتي هي :Area .

من خلال هذا الوصف فإن الخطوات الاساسية هي كما يلي :

أ) ادخال نصف قطر الدائرة .

ب) تعريف ثابت النسبة الثابتة PI

جـــ) حساب المساحة .

د) طباعة النتيجة وهي المساحة .

ولهذا فإن هذا البرنامج سوف يكتب كما يلي :

```
1 PROGRAM Circle (Input,Output) ;

2 CONST Pi=3.14159 ;

3 VAR
      R , AREA : REAL
```

```
4 BEGIN
        READ (R);
        Area := Pi * SQR (R);
        WRITELN (R,Area)
  END.
```

حيث :

1- يمثل عنوان البرنامج .

2 - يمثل تعريف الثوابت التي يجب تعريفها قبل تعريف المتغيرات .

3- يمثل تعريف انواع المتغيرات .

4- يمثل جمل لغة Pascal .

إذن نستطيع القول أن أي برنامج بلغة Pascal يتكون من الاجزاء الرئيسية التالية:

أولا : عنوان البرنامج .

ثانيا : الإعلانات أو التصريحات أو التعريفات والتي تتكون من :

أ- تعريف الثوابت .

ب- تعريف انواع المتغيرات .

ج - تعريف الاقترانات والاجراءات (سنأتي عليها لاحقا)

ثالثا : جمل لغة Pascal

نلاحظ من برنامجنا السابق ما يلي :

1- عنوان البرنامج احتل سطرا واحدا ويتألف من كلمة PROGRAM ثم اسم البرنامج ثم اسم (ملف الادخال واسم ملف الاخراج) .

2- عرفنا أولا الثوابت (CONST) والمتمثلة بها النسبة الثابتة ثم بعد ذلك عرفنا المتغيرات والمتمثلة بنصف القطر ، والمساحة .

3- كتبنا برنامج Pascal محصورا بين جملتي END , BEGIN ووضعنا نقطة بعد جملة END للدلالة على انتهاء البرنامج .

4- استخدمنا الفواصل (الفارزة المنقوطة)(;) بين أي جزئين و لم نستخدمها في الجملة الاخيرة .

5- من الممكن استخدام جمل توضيحية توضع بين نجمتين وقوسين وهذه لا تعتبر جمل Pascal وإنما تعتبر جمل توضيحية ويمكن ان نكتب برنامج Pascal السابق مع جمل توضيحية كما يلي :

```
PROGRAM Circle (Input,Output);
(* this PROGRAM is to Compute AREA of a circle *)          جملة توضيحية

CONST  Pi=3.14159 ;
VAR Radius,Area : REAL ;
(*start the action statement *)                             جملة توضيحية
 BEGIN
(* read the radius *)                                        جملة توضيحية
        READ  (Radius);
        Area := Pi * SQR (Radius);
        (*print out *)                                       جملة توضيحية
        WRITELN (Radius,Area)
END.
```

ولتنفيذ البرنامج السابق باستخدام حاسوب شخصي P. C فإننا نقوم بمتابعة الخطوات التالية :

1. بعد إدخال وطباعة البرنامج وظهوره على الشاشة نستخدم الايعاز Compile الذي يقوم بتدقيق البرنامج حسب قواعد لغة باسكال ثم تكوين ملف الهدف بلغة الالة .

2. بعد ذلك في حالة عدم وجود اية اخطاء نستخدم ايعاز التنفيذ :RUN .

3. ستظهر شاشة الإخراج (الشاشة السوداء) وحركة المؤشر في الزاوية اليسـرى اشارة الى وجوب إدخال مدخلات البرنامج أي قيمة نصف القطر Radius فنقوم بإدخالها من لوحة المفاتيح ثم نضغط مفتاح الإدخال ENTER .

4. يتم تنفيذ البرنامج وتظهر النتيجة النهائية للمساحة .

مع ملاحظة ضرورة التدقيق على البرنامج لعدم حدوث أخطاء حيـث يمكـن ان تكون هناك أخطاء ناتجة عن ترجمة البرنامج أو أخطاء تنفيذية ، أو اخطاء منطقيـة. ومن الأخطاء الشائعة عن ترجمة البرنامج فهي : اخطاء الفواصـل أو النقـاط أو الاقواس أو الإملاء أو استخدام رموز غير مقبوله أو استخدام ارقام تشابه الحروف مثل حرف أو (O) ورقم (0) أو 1 و I ...

اما الاخطاء التنفيذية فهي : اخطاء كتابة في جمل القراءة او الطباعة أو اخطـاء في تعريف المتغيرات أو العمليات الحسابية .

اما الاخطاء المنطقية فهي : اخطاء ناتجة عن التكرار أو استخدام اقترانات بـدون كتابة اسماء ملفاتها في بداية البرنامج .

(6 – 9) جمل التحكم (السيطرة) Control Statement

ان جمل التحكم هي الجمل التي يتم استخدامها في برامج لغـة Pascal لغرض الانتقال من جملة إلى اخرى اي دون تنفيذ البرنـامج بصـورة متسلسـلة (Sequential) ، وذلك عند حاجة البرنامج لتنفيذ تقديم جملة على اخـرى ، أو من الممكن العودة إلى تنفيذ جملة سابقة منفذة (Executed) لغرض استخراج قيم ترتبط بالجملة المنقول لها التنفيذ أو المطلوب العودة لها بالتنفيذ ، وهذا كلـه قبـل الانتقال إلى جمل جديدة تقع ضمن التتابع المنطقي للبرنامج ، وفي بعض الأحيـان يحتاج البرنامج لتكرار تنفيذ مجموعة أو واحدة من الجمل والتعليمات لعدة مـرات. لهذا نستخدم جمل التحكم لمعالجة هذه الامور لتنفيذ عمليات متكررة قد تكـون مشروطة أو غير مشروطة . وقبل أن نستعرض الأساليب المختلفة المستخدمة في لغة

باسكال (PASCAL) للتحكم في سير خطوات البرنامج ، لابد من التطـرق الى مفهوم الجمل المركبة .

(9-6-1) الجمل المركبة The Compound Statement

الجملة المركبة في لغة باسكال تتألف من مجموعة من جمل باسكال البسيطة الـتي يجب تنفيذها بشكل تتابعي والصيغة العامة لها هي كما يلي :

```
BEGIN
            Statement (1);
            Statement (2);
            ..................
            ..................
            ..................
            Statement (n-1);
            Statement (n)
END;
```

ونقصد بجمل باسكال البسيطة هي أي جملة تنتهي بفاصلة منقوطة ، ونلاحظ مـن الصيغة السابقة لجملة باسكال المركبة انها تبدأ بالكلمة المحجوزة BEGIN وتنتـهي بالكلمة END تليها مباشرة فاصلة منقوطة (;) ، ويفصل بين كل جملة بسـيطة وأخرى أيضاً فاصلة منقوطة ما عدا الجملة الأخيرة التي تأتي قبل كلمة END .

(9-6-2) جملة التحكم الشرطية IF – THEN – Statement

تسمى هذه الجملة في بعض الأحيان **بنية اذا ...فإن** ، وهـي الجملـة البنائية IF-THEN-Structure التي تستخدم عند تحقيق شرط محدد أو في بعض الأحيان عند عدم تحقيق شرط معين حيث يتم اختبار جملة الشرط والتي يعبر عنـها (بتعبير منطقي) فإذا تحققت هذه الجملة يتم تنفيذ الجملة التالية لجملة THEN امـا اذا لم تتحقق هذه الجملة فحينها يتم تنفيذ الجملة التي تأتي بعد جملة IF بالتسلسـل والصيغة العامة لهذه الجملة ومنطق هذه الجملة موضحة كما يلي :

IF	Condition	THEN	Statement

حيث Condition هو تعبير منطقي.

Statement جملة باشكال بسيطة او مركبة والصيغة السابقة تعني انـــه اذا تحقق الشرط Condition فيتم الانتقال لتنفيذ جملة Statement التي تأتي بعـــد جملة THEN ، واذا لم يتحقق الشرط فيتم الانتقال إلى الجملة التالية لجملـــة IF الشرطية أي الجمل التي تلي الفاصلة المنقوطة في نهاية جملـــة IF ، والشـــكل (1) يوضح منطق عمل هذه الصيغة .

مثال (33): الايعازات التالية تمثل جملة الشرط بأنواعها وكما هو موضح لكـــل منها :

نوعها	الجملة
جملة شرطية حسابية	1. IF A<10 THEN B:=A+2 ;
جملة شرطية رمزية	2. IF Color < >'Y' THEN Color='G'

```
3. IF(A<=0)OR(B<=12)OR(C<30)THEN

BEGIN
       FLAG:=TRUE;
       WRITELN('INVALID DATA')
END;
```

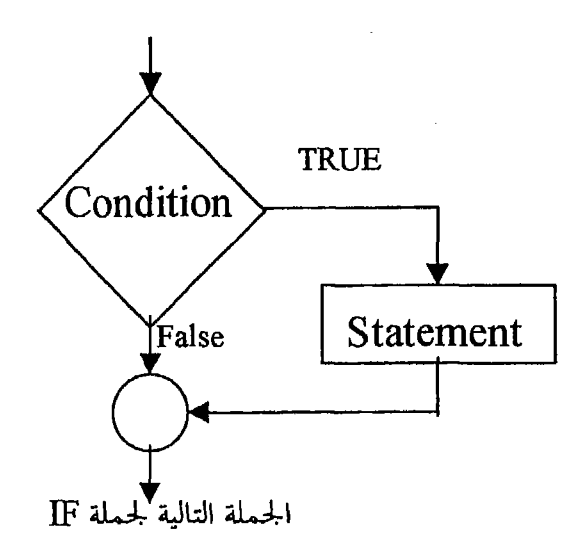
جملة شرطية حسابية تحوي على اكثر من بديل لشرط واحد

```
4. IF (A=200)OR(Sex='F')  THEN  Team := TRUE ;
```

جملة شرطية تحوي على شرط رمزي وشرط حسابي

```
5. IF (B>=0)AND(A='YES') THEN C>=(2*B-1) ;
```

جملة شرطية تحوي على شرط رمزي وشرط حسابي



الشكل(1) يمثل منطق عمل جملة IF-THEN

(9-6-3) جملة التحكم الشرطية

IF...THEN...ELSE... Statement

تسمى هذه الجملة في معناه **إذا ...فإن... وإلا**

وهي الجملة البنائية من جمل الشرط والتي تستخدم عند تحقق الشرط فيتم تنفيذ الجملة الأولى واذا لم يتحقق الشرط فيتم الانتقال إلى الجملة التي تاتي بعد جملة الا ..ELSE وصيغتها العامة على الصورة التالية:

```
IF Condition THEN Statement1 ELSE Statement2;
```

أو يمكن ان تكتب بالشكل التالي :

```
IF condition THEN
      Statement1
ELSE
      Statement2;
```

حيث condition تعبير منطقي Boolean Expression.

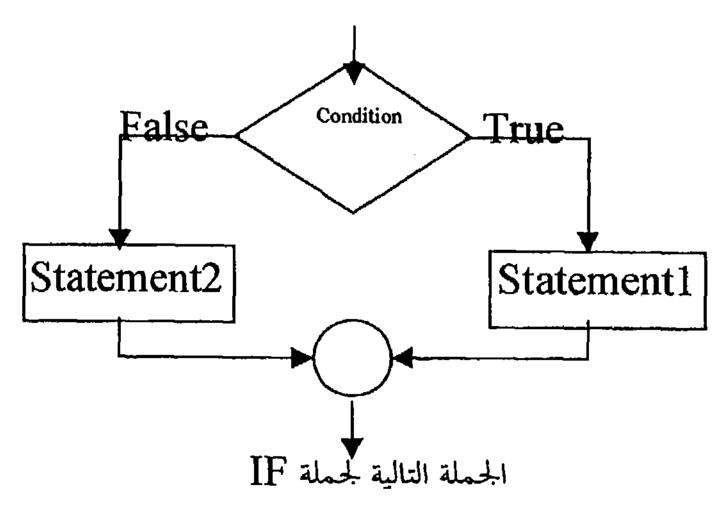
Statement1 جملة من جمل باسكال البسيطة أو المركبة .

Statement2 جملة من جمل باسكال البسيطة أو المركبة .

والصيغة السابقة تعني انه إذا كانت قيمة التعبير المنطقي محققة True فيتم مباشـــرة تنفيذ الجملة التي تأتى بعد جملة Then والتي هي Starement1 ، وإذا كان قيمة التعبير (False) فيتم تنفيذ الجملة التي تأتي بعـــــد جملـــة ELSE والـــتي هـــي Statement2 والشكل (2) يوضح منطق هذه الجملة .

مثال(34):

```
IF  A>= 0  THEN B:=A  ELSE  B:= -A;
WRITELN (A:8:2 , B:8:2);
```



الشكل (2) منطق عمل جملة IF-THEN-ELSE

مثال(35) :

```
IF FORM=SQUR THEN
BEGIN
      READ (Side);
      Area := SQR(Side);
      WRITELN ('Area of Squr =' , Area)
      END
ELSE
      BEGIN
            READLN (Base , Height);
            Area := Base * Hieght/2;
            WRITELN ('Area of Triangle = ' Area);
      End;
End;
```

في المثال أعلاه يتم تنفيذ المقطع الأول الذي يلي جملـة THEN عندمـا يكـون جواب الشرط صحيح وفي حالة عدم صحته يتم تنفيذ الجملة أو المقطع الذي يلـي جملة ELSE

مثال (36) : الجمل التالية مقبولة بلغة باسكال :

```
1- IF  A>=0  THEN  Y:=2*A  ELSE  Y:=12 ;
2- IF FLIAT ='I'THEN B='2'ELSE FIAT='C';
3- IF  A>B  THEN   MAX:=A  ELSE  MAX:=B ;
4- IF  X MOD 2 =0  THEN  B=2  ELSE  MAX:=B ;
5- IF  T=1  THEN
      BEGIN
           READ (H,M);
           Area:=(H*M)/2 ;
       END ;
   ELSE
       BEGIN
            READ (C,D,E);
            S:= (C+D+E)/2;
             Area:=SQRT(S*(S-A)*(S-B)*(S-C)) ;
        END;
   WRITELN(Area);
```

(9-6-4) جملة التحكم الشرطية المتداخلة

Nested IF Statement

ونقصد بجملة التحكم الشرطية المتداخلة ، تداخل جملة IF مع بعضها لاكثر مـن مرة واحدة وهذه الجملة لها عدة اشكال متعددة منها ما يلي :

أ) **الشكل الأول** : والذي يسمى في بعض الأحيان بنية الطبقتين

Two-Layer Nesting

والصيغة العامة لهذا الشكل هو :

```
IF  A1  THEN  IF  A2  THEN Statement1 ELSE Statement2
ELSE IF Conition3 THEN A3 ELSE Statement4 ;
```

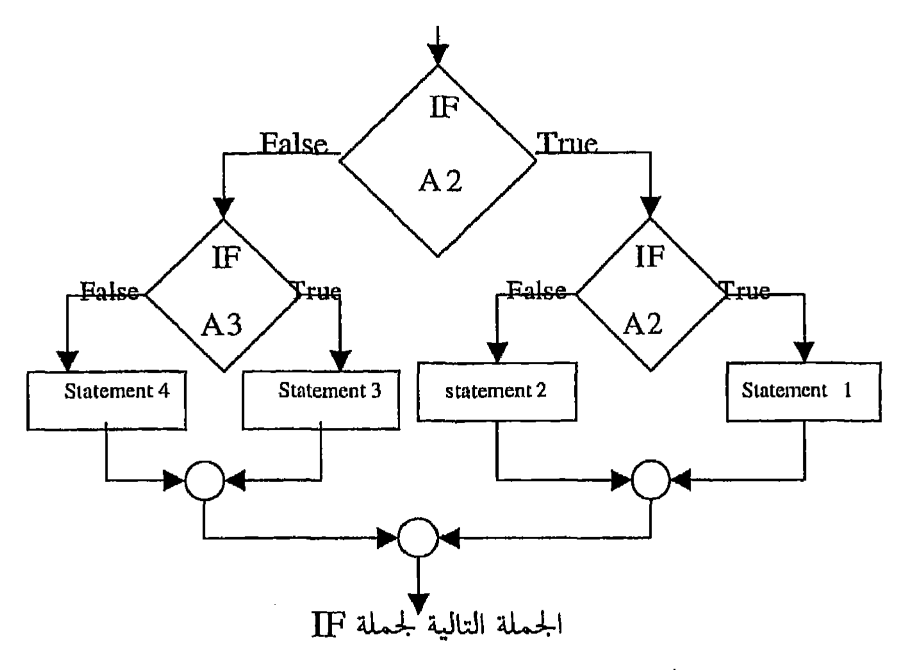
أو بشكل مفصل وأوضح كما يلي :

```
IF A1 THEN
    IF A2 THEN
              Statement1
    ELSE
              Statement2
ELSE
     IF A3 THEN
              Statement3
     ELSE
              Statement4 ;
```

حيث تمثل A3,A2.A1 جملة منطقية ، وتمثل Statement2, Statement1, Statement4 , Statement3 جمل باسكال بسيطة أو مركبة :

وعمل هذه الجملة كما يلي :

يتم اختبار الشرط A1 فإذا كان الشرط متحقق (True) عندئذ يتم تنفيذ جملـــة IF التي تأتي بعد THEN مباشرة (أي اختبار الشرط A2 فإذا كان متحقق يتـــم تنفيذ Statement2 ، أما إذا كان الشرط A1 غير متحقق عندئذ يتم الانتقال الى جملة IF التي تأتي بعد ELSE (أي اختبار الشرط A3 فإذا كان متحقق يتم تنفيذ Statement3 ، وإذا كان غير محقق فيتم تنفيذ Statement4) . ونلاحظ مـــن خلال تركيب هذه الجملة انه يتم تنفيذ فقط إحدى جمل باســـكال البســـيطة أو المركبة ، والشكل (3) يوضح مخطط سير العمليات لهذا النوع من جمل التحكم .

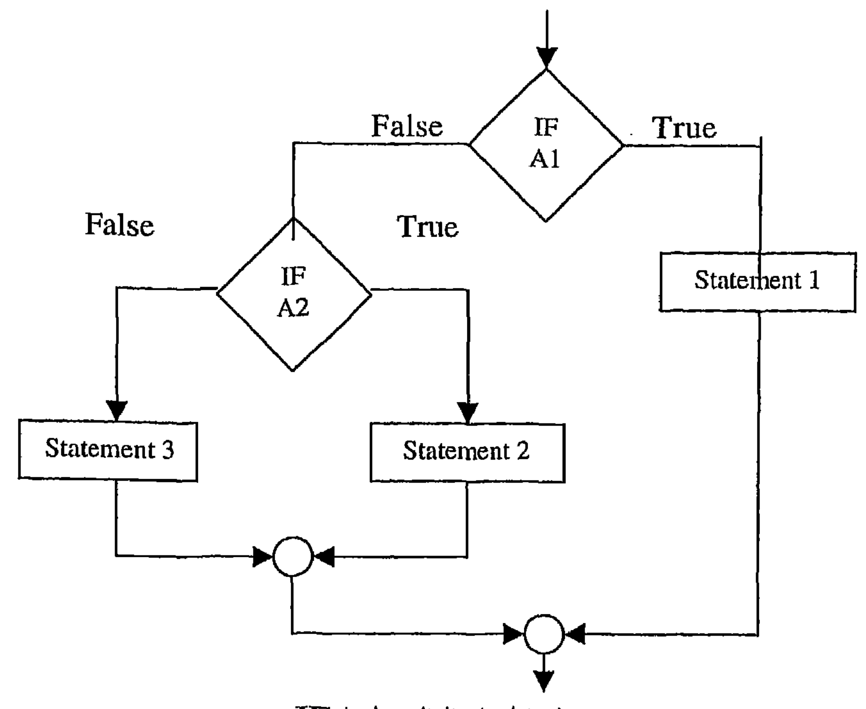


الشكل (3) منطق عمل جملة IF المتداخلة

ب) **الشكل الثاني** : والذي يحوي اكثر من بديل للشرط الثاني والصيغة العامة له:

```
IF A1 THEN
     Statement1
ELSE
     IF A2 THEN
             Statement2
     ELSE
              Statement3;
```

والشكل (4) التالي يوضح مخطط منطقي عمل هذه الصيغة .



الشكل (4) منطق عمل جملة IF المتداخلة

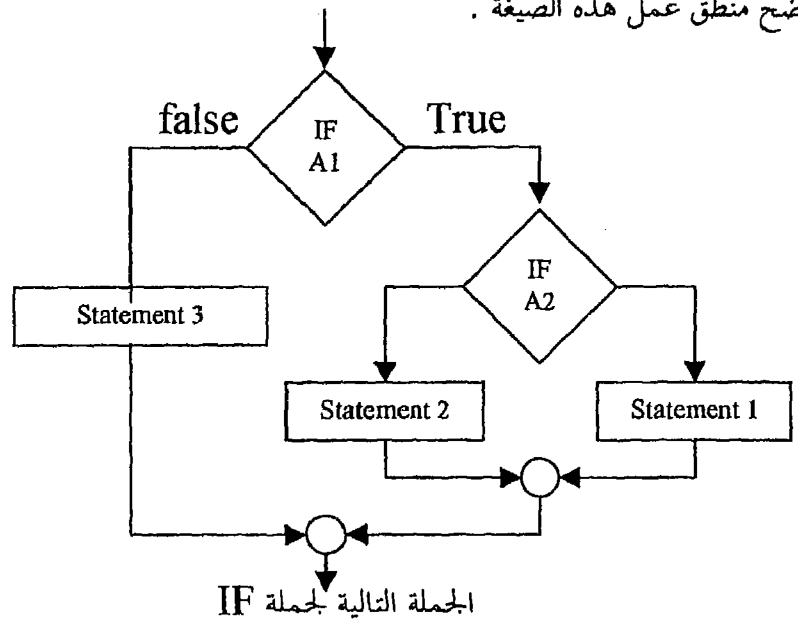
حيث A1.A2 هي تعابير منطقية , Statement1, Statement2 Statement3 هي جملة باسكال بسيطة أو مركبة .

من الشكل (4) و الصيغة السابقة نلاحظ ان التداخل فقط في الجزء الثاني من جملة IF أي بعد جملة ELSE .

ج) **الشكل الثالث** : وهو الذي يتم التداخل فيه في الجزء الأول من جملة IF أي بعد جملة THEN والصيغة العامة لهذا الشكل هو :

```
IF A1 THEN
                IF A2 THEN
                        Statement1
                ELSE
                        Statement2
ELSE
                Statement3;
```

حيث A1 , A2 هـــي تعابـــير منطقيــة ، Statement1 ،Statement2 ، Statement3،والتي تحتل جمل باسكال البسيطة او المركبـــة . والشـــكل (5) يوضح منطق عمل هذه الصيغة .



الشكل (5) يوضح منطق عمل جملة IF المتداخلة

مثال(37) : البرنامج التالي هو لحساب قيمة الاقتران X المعرفة رياضياً حسب القيمة المؤشرة بجانب كل منها :

$$X = \begin{cases} Y^2+1 ; & Y>0 \\ 0 ; & Y=0 \\ -Y+5 ; & Y<0 \end{cases}$$

```
PROGRAM GHAZI (Input,Output);
VAR    X,Y : REAL
BEGIN
        READ (Y);
        IF Y > 0 THEN
                X:=SQR (Y) +1
        ELSE
                IF Y = 0 THEN
                                X:=0.0
                ELSE
                        X:= -Y + 5 ;
        WRITELN ('Y=' Y,'  ', 'X=',X:4:2 );
END.
```

مثال (38) : اكتب برنامج بلغة باسكال يقرأ عدد صحيح ثم يقوم بعملية اختبـــار هذا الرقم ويبين ان كان الرقم فردي أو زوجي .

```
PROGRAM GHAZI (Input,Output);
VAR
      Number , Rem : INTEGER ;
BEGIN
      READ (Number);
      REM:= Number MOD 2;
      IF Rem =0 THEN
           WRITELN('The Number',Number, 'Is EVEN')
      ELSE WRITELN('The Numbeer',Number,' Is  Add')
End.
```

مثال (39) : اكتب برنامج بلغة باسكال يجد جذور المعادلة التربيعية من الدرجـــة الثانية التالية : $AX^2 + BX + C = 0 \quad ; \quad A \neq 0$

كما ان حل هذه المسألة يعتمد على قيمة المميز حيث ان $D = B^2 - 4AC$

فإذا كانت قيمة المميز موجبة فهناك جذرين حقيقين ومختلفين وهما :

$$ROOT1 = \frac{-B + \sqrt{D}}{2A}$$

$$ROOT2 = \frac{-B - \sqrt{D}}{2A}$$

واذا كانت قيمة المميز سالبة فإن للمعادلة جذرين غير حقيقيين

- اذا كانت قيمة المـــيز تســـاوي صفـــر فـــان للمعادلـــة جـــذر مضـــاعف يساوي:

$$ROOT1 = ROOT2 = \frac{-B}{2A}$$

من خلال ما ذكر يمكن تنفيذ البرنامج لايجاد جذور المعادلة التربيعية وكما يلي :

```
PROGRAM ABC (Input,Output);
(*Program calculates the real roots of a quadratic equation*)
VAR
      A,B,C,D            : REAL;
      Root1,Root2  : REAL;
BEGIN
      READ (A,B,C);
      WRITELN('INPUT Data' :) ;
      WRITELN('A=',A:8:2,'B=',B:8:2.'C =',C:8:2);
      IF A = 0.0 THEN
            BEGIN
                  ROOT := - C / B ;
                  WRITELN('Root =',Root : 6 : 2 );
            END;
      ELSE
            D := B * B - 4.0 * A * C ;
      IF  D >= 0.0 THEN
            BEGIN
                  Root1:=(-B+SQRT(D)) / (2.0*A);
                  Root2:=(-B-SQRT(D)) / (2.0*A);
                  WRITLEN (' Root1= ',Root1:6:2);
                  WRITELN (' Root2 = ', Root2:6:2);
            END;
      ELSE
            WRITELN ('No REAL Roots ');
  END.
```

مثال (40) : مخزن تجاري يبيع بضاعة ويعطي خصومات على البضاعة وفق القاعدة التالية :

$$\text{Discount} = \begin{cases} 3\%\ \text{OF S} & ;\ S < 500 \\ 6\%\ \text{OF S} & ;\ 500 <= S < 500 \\ 9\%\ \text{OF S} & ;\ S >= 5000 \end{cases}$$

اكتب برنامج بلغة باسكال لحساب قيم الخصومات المعطاة على مجمل المبيعات

```
PROGRAM GHAZI (INPUT,OUTPUT);
VAR
      S,D : REAL;
BEGIN
      READLN  (S);
      IF S<500 THEN
            D:=0.02*S
      ELSE
            IF S<5000 THEN
                  D:=0.06*S
            ELSE
                  D:=0.09*S;
      WRITELN ('FOR S=' , S:8:4);
      WRITELN ('DISCOUNT= ' ,D:8:4);
END.
```

(9-6-5) جملة التحكم الشرطية **CASE... Statement**

الصيغة العامة لهذه الجملة هي كما يلي :

```
CASE        Expression  OF
            Value 1: statement 1;
            Value 2: statement 2;
            Value 3: statement 3;
            ........................
            ........................
            ........................

            Value n: statement n ;
END.
```

حيث :

Expression – تمثل تعبير حسابي من النوع الصحيح أو يمكن ان يكـون تعبير رمزي أو منطقي .

value – هي قائمة الثوابت و تشمل كل منها قيمة أو مجموعة من القيم تفصل بينها الفارزة .

ومنطق عمل هذه الجملة كما يلي :

يتم اختبار قيمة التعبير Expression ، فإذا كانت value1 تحتوي قيمة هذا التعبير يتم تنفيذ الجملة Statement 1 ويتم الانتقال لتنفيذ الجملة التالية لجملة CASE-END ، وإذا كانت value2 تحوي قيمة هذا التعبير يتم تنفيذ الجملة Statement 2 ويتم الانتقال لتنفيذ الجملة التالية لجملة CASE-END .بمعنى آخر يتم تنفيذ جملة واحدة من مجموعة الجمل المشار اليها في الصيغة العامة حسب وجود قيمة التعبير في احدى value .

أما إذا كانت قيمة التعبير Expression غير موجودة في أي من القيم المحصورة بين value n ....value1 فعند ذلك لا ينفذ الحاسوب أي من الجمل الواردة في بقية جملة حالة الاختيار CASE .

مثال(41) : عند استخدام الاشارة الضوئية للمرور فإن اللون الأحمر هو (للتوقف) واللون الأصفر هو (الانتظار والتهيؤ) واللون الاخضر هو (للتجاوز أو المرور) والمقطع التالي من برنامج يستخدم جملة التحكم الشرطية CASE لحل المشكلة السابقة :

```
CASE color OF
      'Red' :WRITELN ('STOP');
      'Yellow':WRITELN ('WAIT');
      'Green':WRITELN ('PASS');
END;
```

حيث يتم اختيار احدى الجمل اعلاه حسب قيمة اللون COLOR .

مثال (42) : أعطيت علامات الطلبة في جامعة العلوم بالشكل التالي :

التقدير	ادنى علامة
A	90
B	80
C	70
D	60
F	59

أكتب برنامج بلغة PASCAL لقراءة رقم الطالب ومعدله التراكمي وطباعة رقم الطالب وتقديره وفق الجدول اعلاه .

```
PROGRAM prog1 (INPUT,OUTPUT);
VAR
      StudNumber,Grade  : INTEGER ;
      LetterGrade       : Char ;

BEGIN
      READLN (StudNumber , Grade );
      CASE Grade DIV 10  OF
            10 , 9 :LetterGrade :='A';
            8      :LetterGrade :='B';
            7      :LetterGrade :='C';
            6      :LetterGrade :='D';
            5 , 4,
            3 , 2,
            1 , 0  :LetterGrade :='F';
      END;
            WRITELN(Student Number:9,Grade:3,
            LetterGrade);
END.
```

مثال (43) : المقطع التالي من برنامج لاستخدام جملة CASE لقيمة عددية

```
CASE (4*x + 5) OF
      1:Y:=X-Z;
      2:Y:=12;
      4:Y:=-10;
      5:Y:= sqr (x);
      6:Y:=19;
END.
```

مثال (44) :كتابة برنامج يحسب مقدار علاوة الموظف حسب درجته بحيث اذا كانت درجته تساوي :

- 0 ، أو 1 يمنح هذا الموظف علاوة تساوي صفرا .
- 2 ,3 ,4 يمنح هذا الموظف علاوة تساوي 30% من الراتب الاساسي.
- 5 , 6 ,7 يمنح هذا الموظف علاوة تساوي 60% من الراتب الاساسي.
- 8 , 9 يمنح هذا الموظف علاوة تساوي 90% من الراتب الاساسي

أكتب برنامج PASCAL باستخدام جملة CASE لتنفيذ ذلك ؟

```
PROGRAM SALARY (INPUT,OUTPUT);
VAR
      GRADE,BASICSALARY,MONTHSALARY:REAL;
BEGIN
      READ (GRADE , BASICSALARY );
      IF GRADE > 9 THEN
            WRITELN ('INVALID GRADE')
      ELSE
            BEGIN
                  CASE GRADE OF
                  0,1:B:=0 ;
                  2,3,4:B:= 0.3 * BASICSALARY;
                  5,6,7:B:= 0.6 * BASICSALARY;
                  8,9   :B:= 0.9 * BASICSALARY;s
            END;
      MONTHSALARY := BASIC SALARY + B;
      WRITELN('MONTHSALARY=',MONTHSALAR);
      END.
```

(9-6-6) جملة الانتقال GOTO

تستخدم جملة الانتقال GOTO statement لنقـــل تسلســـل تنفيـــذ البرنامج من جملة إلى اخرى عن طريق ا لانتقال غير المتسلسل ، أي الانتقال مـــن التسلسل التتابعي إلى التسلسل المنطقي ، وهذه الجملة لاتستخدم الا في الحـــالات الضرورية لان وجودها في البرنامج يجعله اكـــثر صعوبـــة في القـــراءة والمتابعـــة والتصحيح وصيغة هذه الجملة العامة كما يلي :

```
GOTO        N ;
```

حيث : N يمثل رقم الجملة التي يجب الانتقال إليها ، ويجب أن يكون هذا الرقم معروف في بداية البرنامج باستخدام الكلمة المحجوزة Label .

مثال (45) : البرنامج التالي يوضح استخدام جملة GOTO .

```
PROGRAM GHAZI ( INPUT , OUTPUT );
       LABEL 10 ;
       CONST X = 2.05 ;
       VAR    A,B,C, : REAL ;
BEGIN
       10 READLN ( A,B,C );
       IF A= 0.0 THEN    GOTO     10
       B:= (A + B * C) / A ;
       WRITELN ('B=' , B:10:3) ;
END.
```

(9-7) جمل التكرار المشروط والتكرار باستخدام العدادات :

في الفقرة السابقة قمنا بشرح جمل التحكم التي تقوم بنقـــل ســـير تنفيـــذ العمليات من مكان إلى آخر استناداً إلى حالة الشرط ، وفي تلك الجمل لاحظنـا ان جمل البرنامج تلك كان الحاسوب ينفذها مرة واحدة .

في موضوعنا الجديد الحالي باستخدام جمل التكرار المشروط أو التكـــرار باستخدام العددات فإن جمل يتم تنفيذها لاكثر من مرة واحدة ، ويمكن التمييز بين

الانواع التالية من جمل التكرار :

1- استخدام جملة التكرار المشروط WHILE-DO
2- استخدام جملة التكرار المشروط REPEAT-UNTIL
3- استخدام جملة التكرار المتعدد FOR-NEXT
4- استخدام جملة التكرار المتداخل (Nested Loop Structure)

وسوف نقوم بشرح كل من تلك الانواع بشكل مفصل .

(9-7-1) استخدام جملة التكرار المشروط WHILE – DO :

أ) جملة التكرار المشروط WHILE-DO البسيطة :

في هذه الجملة نلاحظ ان عدد التكرارات التي تحدث لا يكون معروفاً مسبقاً وذلك لارتباط هذه الجملة بتحقق أو عدم تحقق الشرط فإذا كانت قيمة الشرط صح True فسوف تستمر عملية التكرار إلى ان يصبح قيمة الشرط خطأ False ، فعندها يتم الانتقال إلى الجملة التالية لجملة WHILE وبنية هذه الجملة كما يلي:

WHILE	Condition	DO	Statement ;

وهذه الجملة تعني انه مادام الشرط محقق فسيتم تنفيذ الجملة Statement الموجودة بعد كلمة DO ، ويستمر هذا التكرار إلى ان ينعدم تحقق الشرط ، فيتوقف التكرار ، وهذا النوع من الجمل التكرارية يدعى بالجمل التكرارية البسيطة والمثال التالي يوضح تركيب هذه الجملة :

مثال (46) :

```
..................
..................
Nmber:=0;
WHILE Number <=1000 DO
      Number:=Number+1;
..................
..................
```

نلاحظ في هذا المقطع من برنامج ما أن المتغير Number اخـــذ القيمـــة صفر ، وبعد عبارة WHILE وضعنا الشرط الذي يعني ما دامت قيمة المتغير أقـل أو تساوي (1000) فان الحاسوب سينفذ الجملة التي تلي عبارة DO أي سـتصبح قيمة Number تساوي (1) ، ويعود إلى اختبار قيمة Number في الشرط وبمـا أنها اقل من (1000) فإنه سيقوم بتكرار تنفيذ الجملة التي تلـــي DO ، وهكـــذا يستمر بالتكرار إلى أن تصبح قيمة Number تساوي إلى (1001) عندئذ يتوقف الحاسوب عن التكرار ويتابع تنفيذ بقية جمل البرنامج التي تلي الفاصلة المنقوطة .

ب) جملة التكرار المشروط WHILE-DO المركبة :

في بعض الأحيان يتم استخدام جملة التكرار المشـــروط WHILE-DO بشكل مركب ، أي ان جملة حلقة التكرار تتكون من اكثر من جملة واحدة والبنية العامة لهذه الجملة كما يلي :

```
WHILE  Condition  DO
BEGIN
        Statement1;
        Statement2;
        Statement3;
        ...............
        ...............
        Statement n.-1
        Statement n
END;
```

وهذه الجملة تعني أنه يتم أولاً اختبار الشرط (المنطقي) فإذا تحقـــق هـــذا الشرط أي اصبح TRUE يتم تنفيذ جملة حلقة التكرار (المركبة) اما اذا لم يتحقـق الشرط فيتم الانتقال في التنفيذ إلى الجملة التالية لجملة WHILE وهكذا. والمثـال التالي يوضح ما سبق :

مثال (47) : اكتب مقطع برنامج باسكال لحساب قيمة Y المعرفة رياضياً علـــى الشكل التالي $Y=4X^2+X+7$ ولجميع قيم X المحصورة بـين (2-,2) أي $-2<=X<=2$ وبزيادة مقدارها 0.5 .

```
...............
...............
X:=-2.0;
WHILE X<=2.0 DO
BEGIN
      Y:=4*SQR(X) + X + 7.0;
      WRITELN ( X:5:2,'  ',Y:5:2);
      X:= X + 0.5
END;
............
............
```

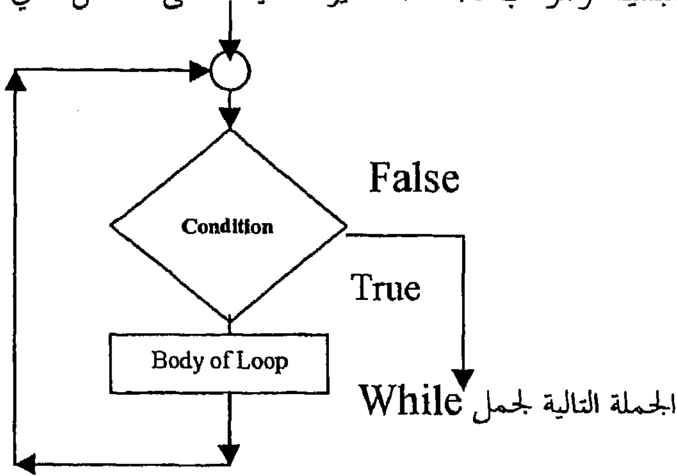
نلاحظ في هذا المقطع من البرنامج ان X= - 2 في البداية ، وبعد عبارة WHILE وضعنا الشرط الذي يعني انه مادامت قيمة X اقل أو تساوي 2 فـإن الحاسـوب سينفذ مجموعة الجمل التي تلي عبارة DO والمحصورة بين عبارة BEGIN وعبـارة END ، أي ان الحاسوب سيقوم بحساب قيمة Y حسب العلاقة المعطاه والـتي تساوي (21) ثم يطبع على شاشة المخرجات ما يلي :

2.00		21.00
قيمة X	فراغ	قيمة Y

ثم يحسب قيمة X الجديدة التي تساوي 2.5 ، وعندما يصل إلى عبارة END يعود إلى اختبار الشرط الذي يلي WHILE وبما انه متحقق يعود الحاسـوب لينفـذ الجمل بعد DO كما في المرة السابقة مع اخذ قيمة X الجديدة التي تسـاوي 2.5 وهكذا ... ويتوقف التكرار عندما تصبح قيمة X تساوي 2.5 وعندهـا يتـابع الحاسوب تنفيذ الجمل التي تلي ;END

بناءاً على ما سبق نستطيع ان نعبر عن جملـة WHILE-DO بنوعيـها

البسيط والمركب ، بمخطط سير العمليات على الشكل التالي :



الشكل (6) منطق عمل جملة WHIL-DO

من هذا الشكل نلاحظ انه اذا كانت قيمـــة الشـــرط Condition هـــي False فيتم الانتقال إلى الجملة التالية لجملة WHILE ، واذا كانت قيمة الشرط Condition هي TRUE فيتم الانتقال لتنفيذ جملة التكرار Body of Loop بنوعيها البسيط والمركب .

مثال(48) : اكتب برنامج بلغة باسكال لايجاد الوسط الحسابي لمجموعة من القيـم المعلومة .

الحل : الوسط الحسابي نرمز له Average يحسب من العلاقة التالية :

$$AVERAGE = \frac{SUM}{N}$$

و SUM تمثل مجموعة الأعداد والتي تبدأ X1 ، وتنتهي في X n .

أي ان :

$$AVERAGE= \frac{SUM}{N} = \frac{X1+X2+X3+X4+\ldots+Xn}{N}$$

البرنامج :

```
PROGRAM Exam1 (Input,Output);
VAR
      N, Count              : INTEGER;
      SUM, X, AVERAGE: REAL;
BEGIN
      WRITELN ('Enter Number of Value : ');
      READLN(n);
      Sum:=0.0;
      Count:=1;
      WRITELN ('Enter Values : ');
      WHILE count <=n DO
      BEGIN
            READ (x);
            Sum:=sum+x;
            Count:=count+1;
      END;
      AVERAGE:=SUM / N;
      WRITELN ( THE AVERAGE=',AVEARAGE: 5:2);
END.
```

و عند تنفيذ البرنامج ستبدو شاشة المخرجات وكما يلي:

```
Enter  number of value :5
Enter Value :
40
50
25
10
The AVERAGE = 30.00
```

مثال(49) : اكتب برنامج بلغة باسكال لايجاد مضروب عدد حسب القـــاعدة التالية :

$$N! = \begin{cases} 1 & ; N = 0 \\ 1*2*3*.....*N & ; N \neq 0 \end{cases}$$

```
PROGRAM EXAM2 (Input,output);
VAR
      FACT,COUNT,N : INTEGER ;
BEGIN
      READLN (N) ;
      FACT:=1;
      COUNT:=1;
      IF N <>0 THEN
            WHILE COUNT<=N DO
            BEGIN
                  FACT:=FACT*COUNT;
                  COUNT:=COUNT +1;
            END;
      WRITELN ('FACTORIAL OF',N,'=',FACT :5);
      READLN;
END.
```

فإذا أدخلنا قيمة N تساوي 5 عندئذ ستبدو شاشة المخرجات كما يلي :

```
5
FACTORIAL OF 5 = 120
```

مثال (50) : اكتب برنامج بلغة باسكال لايجاد القيمة الصغرى من بين مجموعـــة قيم عددية صحيحة :

```
PROGRAM EXAM3 (Input,output);
VAR
      N,NUMBER,MIN,COUNT : INTEGER;
```

```
BEGIN
        MIN:=MAXINT;
        COUNT:=1;
        WRITELN ('Enter Number of Integer :');
        READLN(N);
        WRITELN ('Enter Values :');
        WHILE COUNT <=N DO
        BEGIN
                READLN (NUMBER);
                COUNT:=COUNT + 1;
                IF NUMBER < MIN THEN MIN:=X;
        END;
        WRITELN ('The Minimum  Value =',MIN);
        READLN;
END.
```

عند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```
Enter Number of INTEGER : 4
Enter Values :
78
45
30
85
the Minimum Value = 30
```

(9-7-2) جملة التكرار المشروط Repeat-Until

أ) جملة التكرار المشروط Repeat-Until البسيطة :

ان جملة التكرار المشروط Repeat-Until شبيهة بجملة التكرار المشروط Whele-Do حيث ان عدد التكرارات التي تحدث فيها لا يكون معروفاً وذلـــك لارتباط هذه الجملة بتحقق أو عدم تحقق الشرط الذي يتبعها ، فمنطق هذه الجملة بأنه يتم تنفيذ جملة التكرار ثم يتم اختبار الشرط فاذا لم يتحقق الشـــرط وقيمتــه False يستمر التكرار إلى أن يتحقق الشرط بحيث يصبــح True عندهــا يتــم

الانتقال إلى تنفيذ الجملة التالية لجملة Repeat والبنية العامة لهذه الجملة هي كما يلي :

REPEAT	Statement	UNTIL	Condition;

يختلف تركيب هذه الجملة عن جملة WHELE-DO في انه يتم تنفيذ الجملة Statement على الاقل مرة واحدة ثم يتم اختبار تحقق الشرط بينما كما لاحظنا في الفقرة السابقة ان جملة WHELE-DO يتم اختبار تحقق الشرط أولاً ثم يتم تنفيذ الجملة Statement التي تأتي بعد DO . والمثال التالي يوضح تركيب هذه الجملة:

مثال (51) :

```
.............
.............
number :=0;
REPEAT
        number :=number+1 ;
UNTIL number > 1000;
.............
.............
```

نلاحظ في هذا المقطع من البرنامج أن المتغير number اخذ القيمة صفر في البداية، وبعد عبارة Repeat التي تعني كرر ، سينفذ الجملة التالية :

number := number +1

والتي ستصبح قيمة number تساوي (1) وبعد عبارة Until سيقوم بالإختبار الذي يعني انه كرر الجملة السابقة حتى تصبح قيمة number اكبر من (1000)، وبما ان قيمة number تساوي (1) واقل من (1000) عندئذ سيقوم الحاسوب بتكرار جملة (number := number + 1) والتي ستصبح قيمة number تساوي (2)، وبعد عبارة Until سيتم اختيار الشرط وهكذا ....، إلى ان تصبح

قيمة المتغير number اكبر تماماً من 1000 وعندئذ سيتابع تنفيذ الجمل التي تلـــي الفاصلة المنقوطة .

ب) استخدام جملة التكرار المشروط Repeat-Until (المركبة)

ومعنى هذه الجملة التكرارية المشروطة والمركبة هو انه يتم القيام بتكـــرار مجموعة من الجمل عدداً من المرات إلى ان يتحقق الشرط Condition ويكـــون جوابه True عندها يتوقف التكرار والبنية العامة لهذه الجملة هي كما يلي :

```
REPEAT
          Statement 1;
          Statement 2;
          Statement 3;
          Statement n;
UNTIL Condition ;
```

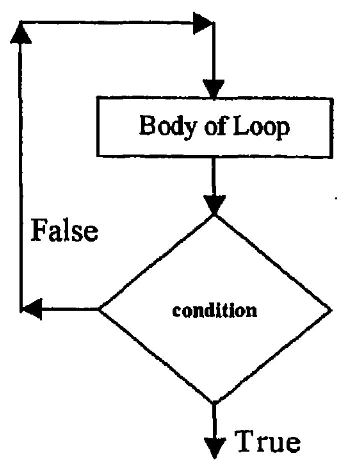
مثال (52): اكتب مقطع برنامج باسكال ينفذ الجملـة WRITELN (Count) تسعة مرات ، حيث يأخذ العداد Count القيـــم 9,8,7,6,5,4,3,2,1 علـى التوالي.

```
..........
..........
count :=1;
REPEAT
      WRITELN (count);
      count:=count +1;
UNTIL count > 9;
..........
..........
```

نلاحظ في هذا المثال ان العداد count يأخذ القيمة (1) ويزداد قيمته أيضا بمقدار واحد (1) في كل مرة حتى يتحقق الشرط فيصبح العداد count يساوي أو أكـبر من 9 فيتوقف عمل التكرار .

وبناء على ما سبق نستطيع أن نعبر عن جملة REPEAT ....UNTEL

البسيطة والمركبة بمخطط سير العمليات التالي:



الى الجملة التالية لجملة REPEAT

الشكل (7) يمثل منطق عمل جملة (REPEAT-UNTIL)

حيث نلاحظ من هذا الشكل (7) أنه يتم الاختبار للجملة بعد تنفيذ جملة حلقة التكرار وهذا يعني أن جملة Repeat ينفذ التكرار مرة واحدة على الاقل قبل الانتقال إلى الجمل التالية في البرنامج .

مثال (53) : أعد حل المثال (48) باستخدام جملة التكرار المشروط REPET-UNTIL

```
PROGRAM EXAM5 ( INPUT ,OUTPUT )
VAR
      N,count : INTEGER;
      sum,AVERGE :REAL;
BEGEN
      sum:=0.0;
      count:=1;
      WRITELN ('Enter Number of value :');
      READLN (N);
      WRITELN ('Enter Values:');
      REPEAT
            READLN (X);
```

```
            sum:=sum + x ;
            count:= count + 1;
      Until count > N ;
      AVERAGE := sum / N ;
      WRITELN ('THE AVERAGE=',AVERAGE :5:2);
END.
```

عند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```
Enter Number of Value :5
Enter Values :
40
50
25
25
10
THE AVERAGE = 30.00
```

مثال (54) : أعد حل المثال (49) باستخدام جملة التكرار المشروط -REPEAT . UNTIL

```
PROGRAM EXAM6 (INPUT.OUTPUT);
VAR
      FACT,COUNT,N: INTEGER ;
      BEGIN
            WRITELN ('Enter Number:');
            READLN (N);
            FACT:=1;
            COUNT:=1;
            IF N <> 0 THEN
                  REPET
                        FACT:= FACT * COUNT ;
                        COUNT:= COUNT + 1;
                        UNTEL COUNT > N ;
      WRITELN ('FACTORIAL of ',N,'=',FACT :5);
      READLN;
END.
```

وعند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```
Enter number :5
FACTORIAL of 5 =   120
```

مثال (55) : اكتب برنامج بلغة باسكال لايجاد اكبر قيمة من قائمة تحوي على N قيمة عددية صحيحة .

اذن المطلوب في هذا البرنامج هو ايجاد القيمة العظمى Maxint .

البرنامج :

```
PROGRAM EXAM7(Input,Output);
VAR
      N,X,largest,count : INTEGER ;
BEGIN
      largest:=-MAXINT;
      count:=0;
      WRITELN ('Enter Number of Value :');
      READLN('N');
      WRITELN('Enter Value :');
      REPEAT
            READ (X);
            count:=count +1;
            IF X > largest  THEN  largest :=X;
      UNTIL count=N;
      WRITELN ('The Largest Value=',largest);
      READLN;
END.
```

وعند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```
Enter Number of Value 5
Enter Values :
65
87
98
0
94
The Largest Value = 98
```

(3-7-9) جملة التكرار FOR

نستخدم هذه الجملة لتكرار تنفيذ مجموعة من جمل باسكال عدداً معروفاً ومحدداً من المرات ، والبنية العامة لهذه الجملة :

```
FOR Counter := Initial Value  TO Final Value DO Statement;
```

أو

```
FOR Counter:=Initial Value DOWNTO Final Value DO
Statement;
```

حيث

Counter	اسم العداد لمتغير من نوع صحيح
Initial Value	القيمة الإبتدائية للعداد
Final Value	القيمة النهائية للعداد
To	تستخدم في حالة العد التصاعدي
	اي عندما Initial Value < Final Value
DOWNTO	تستخدم في حالة العد التنازلي
	اي عندما Initial Value > Final Value
Statement	الجملة التي تحتاج إلى التكرار

ملاحظة : الجملة Statement التي تحتاج إلى التكرار عدداً من المرات يمكن ان تكون بسيطة (أي جملة واحدة) ويمكن ان تكون مركبة (عدة جمل) وفي هذه الحالة يجب أن تبدأ الجملة المركبة بعبارة BEGIN وتنتهي الجملة بعبارة ;END أي تأخذ البنية العامة الشكل التالي :

```
FOR Counter := initial value TO final value DO
BEGIN
        Statement 1;
        Statement 2;
        :
        :
        Statement n;
END;
```

مثال (56) : اكتب مقطع برنامج باسكال يجمع قيم الأعداد من 1-9 .

في حالة استخدام جملة TO التصاعدية يكون المثال كما يلي :

```
.........
...........
SUM := 0;
FOR COUNT := 1 TO 9 DO
SUM := SUM + COUNT;
..............
.............
```

وفي حالة استخدام جملة DOWNTO التنازلية فيصبح المثال السابق كما يلي :

```
.........
........
SUM:=9;
FOR COUNT:=9 DOWNTO 1 DO
SUM:=SUM+ COUNT;
........
...........
```

وعمل هذا الجزء من البرنامج كما يلي :

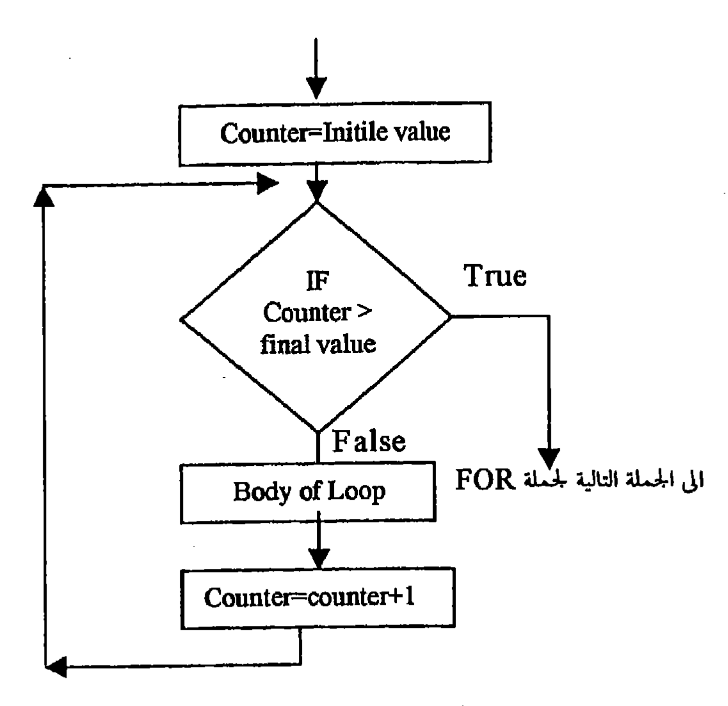
- يبدأ عمل العداد بمقدار القيمة الابتدائية المعرفة في جملة التكرار .
- يتم مقارنه القيمة الجديدة للعداد بالقيمة النهائية فإذا كانت القيمتين متساويتين يتم التوقف واذا كانت قيمة العداد الأولية اقل من النهائية يتم تكرار تنفيذ الجملة التي تأتي بعد عبارة DO علماً أن قيمة العداد في حالة استخدام القيمة التصاعدية يتم إضافة واحد (1) وفي حالة استخدام القيمة التنازلية يتم إنقاص (1) وهكذا .

والشكل (8) يمثل منطق عمل الجملة FOR

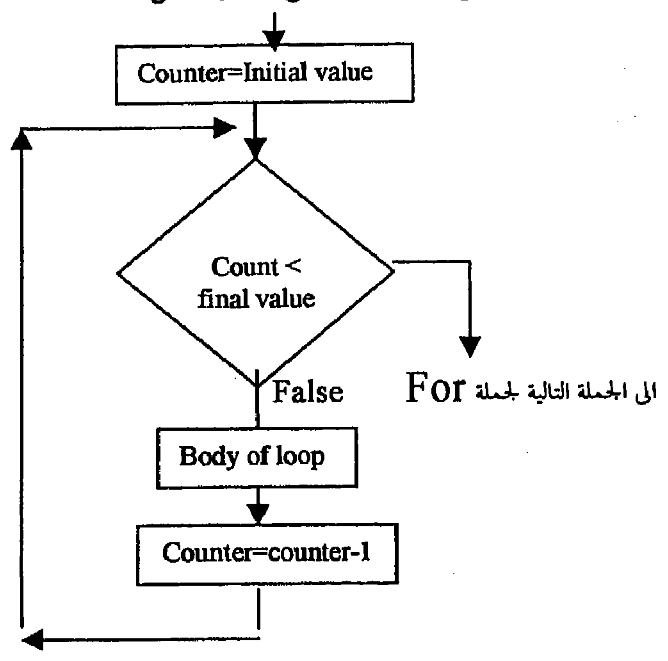
في حالة : Initial Value < Final Value أي ان القيمة الابتدائية للعداد اصغر من القيمة النهائية .

والشكل (9) يمثل منطق عمل الجملة For

في حالة : Initial Value > Final Value أي القيمة النهائية اكبر من القيمة النهائية .

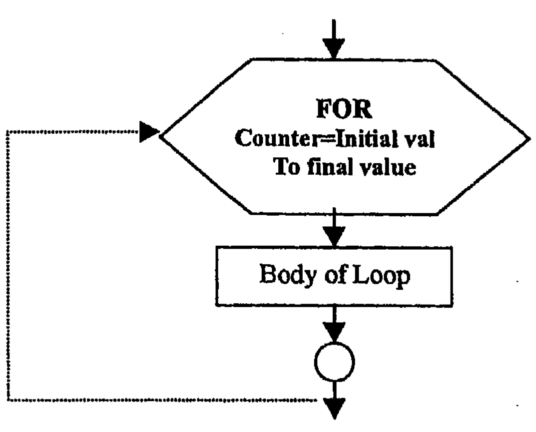


والشكل (8) اعلاه يمثل منطق عمل FOR-TO



الشكل (9) منطق عمل جملة FOR-DOWNTO

وعند رسم مخططات سير العمليات يعبر عن جملة FOR كما هـــــو موضـــح في الشكل (10) :



الى الجملة التالية لجملة FOR

الشكل (10) مخطط سير العمليات لجملة FOR

مثال(57) : اعد حل المثال (48) السابق باستخدام جملة التكرار FOR .

```
PROGRAM EXAM9(Input,Output);
VAR
      Count,N : INTEGER
      Sum,Average,X : REAL;
BEGIN
      WRITELN ('Enter Number of Value:');
      READLN (N);
      Sum := 0.0;
      WRITELN('Enter Values :');
      FOR count := 1 TO n Do
      BEGIN
            READLN(X);
            Sum:=Sum+1;
      END;
      Average := Sum/N;
      WRITELN('The  Average = ', Average :5:2);
END.
```

وعند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```
Enter Number of Value : 5
Enter Values
40
50
25
25
10
The Average = 30.00
```

مثال (58) : اعد حل المثال (49) السابق باستخدام جملة التكرار FOR

```
PROGRAM EXAM10(Input,Output);
VAR
      FACT,COUNT,N : INTEGER;
BEGIN
      WRITELN('Enter The Number :');
      READLN (N);
      FACT:=1;
      FOR COUNT:=1 TO N DO
      BEGIN
            FACT := FACT * COUNT;
      END;
      WRITELN ('FACTORIAL Of ',N,' ' ,FACT :5);
      READLN;
END.
```

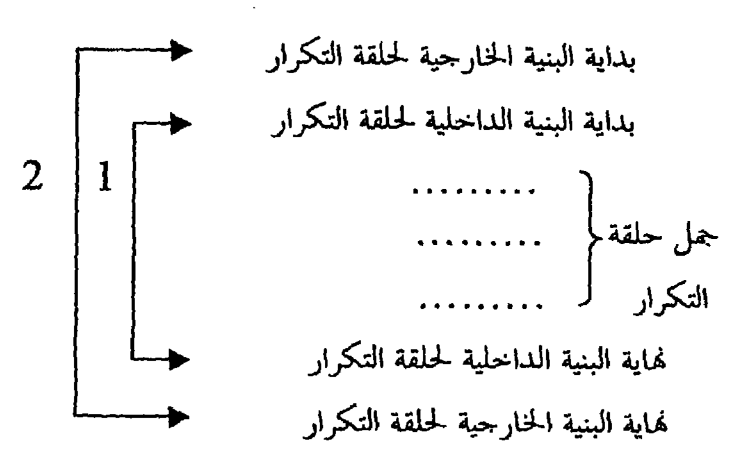
وعند تنفيذ البرنامج ستبدو شاشة المخرجات كما يلي :

```
Enter The Number : 6
FACTORIAL Of 6 = 720
```

(4-7-9) جمل التكرار المتداخلة Nested Repetition Structure

نستخدم هذه الجملة في الحالات التي يحوي فيها البرنامج على اكثر من حلقة تكرارية متداخلة أي اننا نقوم بوضع بنية تكرارية داخل بنية تكرارية اخرى

وهذا يؤدي إلى تكرار عمليات مكررة سابقة شريطة ان تكون الجملة التكرارية الثانية واقعة داخل بنية جملة التكرار الخارجية وبتفصيل اكثر نوضح هذا التكرار من خلال المخطط في الشكل (11) :



والشكل (11) يمثل منطق عمل جملة FOR المتداخلة

ولنوضح ما سبق من خلال المثالين التاليين الأول لكتابة برنامج بدون تداخل والثاني باستخدام تداخل جمل التكرار .

ملاحظة : ان التداخل يمكن ان يتم باستخدام جملة التكرار WHILE-DO أو جملة التكرار REPEAT-UNTIL أو جملة التكرار FOR أو خليط بينهما وهذا يتعلق بالمبرمج أولاً وبنوع خوارزمية الحل ثانياً .

مثال (59) : اكتب برنامج بلغة باسكال لحساب معدل اعمار طلبة احدى الشعب في جامعة معينة .

الحل : نفرض ان عدد الطلاب هو 60 طالب .

اذن المطلوب قراءة اعمار 60 طالب ثم يتم جمعها في Counter ويطلق عليه اسم المجموع sum ، ثم تتوقف العملية بعد قراءة الـ(60) عمر وخزنها في sum ، ثم يتم ...من حلقة التكرار ، ثم يتم تقسيم مجموع

الاعمار الـــ(60) على عدد الطلاب (60) وذلك لحساب الوسط الحسابي (المعدل) للعمر ثم يتم طبع قيمته وتتوقف .

نرمز لعمر الطالب بـــ A ، ومجموع اعمار الطلبة في الشعبة بــــsum، ومعدل العمر بـــ Mean

```
PROGRAM EXAM11(Input,Output);
VAR
      COUNT : INTEGER;
      A,SUM,MEAN:REAL;
BEGIN
      SUM:=0.0;
    ┌►FOR COUNT:=1 TO 60  DO
تمثل │     BEGIN
حلقة │     READ (A);
التكرار│    SUM:=SUM+A;
    └►END;
      MEAN:=SUM/60;
      WRITELN ('Mean =' , MEAN);
END.
```

مثال (60) : اكتب برنامج بلغة باسكال لحساب معدل اعمار الطلبة في كل شعبة من جامعة ما مع العلم ان الجامعة تضم عشرة شعب وكل شعبه تضم 60 طالباً .

الحل : ان هذا البرنامج هو تعميم للبرنامج المعروض في المثال (11) ، إذا اننا سنقوم بتكرار حلقة التكرار في المثال (11) عشرة مرات ، فإذا رمزنا بـــCLASS للعداد الذي سيستخدم لعد الصفوف العشرة ، فإن البرنامج يأخذ الشكل التالي :

```
PROGRAM EXAM12(Input,Output);
VAR
        CLASS,COUNT:INTEGER;
        A,SUM,MEAN : REAL;
```

```
BEGIN
FOR CALSS:=1 TO 10 DO
     BEGIN
        SUM:=0.0;
        FOR COUNT :=1 TO 60 DO
        BEGIN
            READ (A);
            SUM :=SUM+A;
        END;
        MEAN:=SUM/60;
        WRITELN('Class=',CLASS,'Mean=',MEAN);
     END;
END.
```

البنية الخارجية 10مرات

البنية الداخلية 60مرة

مثال (61) : اعد كتابة البرنامج في المثال (60) باستخدام REPET-UNTIL

```
PROGRAM EXAM13(Input,Output);
VAR
     CLASS,COUNT :INTEGER;
     A,SUM,MEAN : REAL;
BEGIN
     CLASS=1;
     REPEAT
        COUNT := 1;
        SUM := 0.0;
        REPEAT
            READ(A);
            SUM:=SUM+A;
            COUNT := COUNT + 1;
        UNTIL COUNT > 60;
        MEAN:=SUM/60;
        WRITELN('Class=',CLASS,'Mean=',MEAN);
        C LASS=CLASS+1;
     UNTIL CLASS > 10 ;
END.
```

مثال (62) : اعد كتابة البرنامج في المثال (12) باستخدام WHILE-DO (يترك للطالب) .

مثال (63) : اكتب برنامج بلغة باسكال لحساب قيمة العدد النيبري e حسب العلاقة التالية :

$$e = 1 + \frac{1}{2!} + \frac{1}{3!} \qquad \frac{1}{N!}$$

مستخدما التكرار المشروط WHIL-DO لحلقة التكرار الخارجية وجملة FOR لحلقة التكرار الداخلية .

الحل : اسماء المتغيرات التي ستستخدم في البرنامج هي :

N عدد الحدود

COUNT1 عداد لعد حدود العلاقة

COUNT2 عداد لاستخدامه عند حساب مضروب الأعداد في المقامات

SUM مجمع لتجميع حدود العلاقة

FACT لوضع فيه مضروب عدد

```
PROGRAM EXAM15 (Input,Output);
VAR
      N,CODUNT1,COUNT2,FACT :INTEGER;
      SUM :REAL;
BEGIN
       WRAITELN(Enter THE Number of Terems:');
      READLN (N);
      COUNT1 :=1;
      SUM:=0.0;
      WHILE COUNT1 <=N DO
      BEGIN
            FACT := 1;
            FOR COUNT2 :=1 TO COUNT1 DO
                  FACT:=FACT*COUNT2;
```

```
            SUM := SUM + 1.0 / FACT;
            COUNT1:= COUNT1 + 1;
      END;
      WRITELN ('e=',SUM:4:2);
END.
```

وعند تنفيذ البرنامج تبدو شاشة المخرجات كما يلي :

```
Enter The Number of Terms : 5
E = 1.72
```

مثال (64) : اكتب برنامج بلغة باسكال لطباعة الحرف X على خمسة اسطر بالشكل التالي :

في السطر الأول حرف واحد

في السطر الثاني حرفان

في السطر الثالث ثلاثة احرف

في السطر الرابع اربعة احرف

في السطر الخامس خمسة احرف

```
PROGRAM EXAM16 (Input,Output);
VAR
      ROW,LITER : INTEGER;
BEGIN
      FOR ROW := 1 TO 5 ROW
      BEGIN
            FOR LETTER:=1 TO ROW DO
WRITELN('X');
            WRITELN;
      END;
END.
```

وعند تنفيذ البرنامج ستبدو على الشاشة كما يلي :

```
X
XX
XXX
XXXX
XXXXX
```

تمارين وأسئلة عامة

1- إفرض ان

```
VAR
    A,B,C : INTEGER ;
    X,Y,Z : REAL ;
```

وضح أي من الجمل التالية مقبولة أو غير مقبولة في لغة باسكال كجمل اسناد .

```
1. X:=B*(C+Y)/Y
2. B:=(A+C) DIV 2
3. A:=((3*B) MOD (C DIV 2)) * 20.0
4. A=TRUNC (((4*C) MOD (C DIV 2) + 10.2))
5. Y:=X*X+X+3*3*SQR(Y)
```

2- اكتب الجمل اللازمة للاعلان عن نوع كل من المتغيرات التالية :

المتغير	نوعه
1. ROOT	حقيقي
2. X	حقيقي
3. Count	صحيح
4. Color	رمزي
5. State	منطقي
6. Index	صحيح

3- إفترض ان قيمة a هي الصواب True

b هي الخطأ False

c هي الصواب True

جد قيمة كل من التعابير المنطقية التالية

```
1. a AND ( b OR c )
2. ( a AND b ) OR ( a AND c )
3. NOT a OR NOT c
4. a AND NOT b OR NOT c
```

4- جد قيمة التعابير الحسابية التالية :

1. Round (-2.8)
2. Round (-5.1)
3. Trunc (-2.8)
4. Trunc (-5.1)
5. Trunc (3.9)
6. Round (3.9)
7. ABC (3)
8. SQR (-2)

5- حدد نوع التعبير لكل مما يلي :

1. 6*3+14/2
2. Counter
3. SQR(Z+4)/6
4. Minimum=200
5. Root<=2.0

6- اكتب التعابير التالية بلغة باسكال :

1. (Z+A) (B+A)
2. 6(M-Y)
3. QRZ^2+5YX
4. $X+CY^{-2}$
5. $Z^3 / 3!$

7- أكتب التعابير التالية بلغة باسكال بالاستعانه بالاقترانات المكتبية القياسية :

1. | C+D | GM
2. $\overline{A^2 + C^2}$
3. $\sqrt{COS(A+B)}$
4. $TAN^{-1} (C/D)$
5. E^{c+b}

8- اكتب المعادلات التالية بلغة باسكال

1. $G=(C^2 / 2!) + (C^3 / 4!)$
2. $H= (X^8R-ST^4) / (aX+b)$
3. $D= (1+L)^{1/2}$
4. $L:= \sqrt{X} - 4X+5) / (4(X-1))$
5. $M:=(R\, e^{9Ln(X)} - St^4) / (a(x+b))$

9- نفرض ان :

```
VAR
      A,B,C : REAL;
      I,J,K : INTEGER;
```

وبفرض ان سجلات ملف البيانات المدخلة هي :

```
44     5.56 16    14
5      17         21
19     11
```

والمطلوب ما هي نتائج تنفيذ جمل القراءة التالية :

```
READ(I,A,B);
READLN( C );
READLN( K ) ;
READ (J);
```

10- بفرض ان :

```
X=5.6,Y=2        ,Z=3.0
```

فما هي نتائج تنفيذ الحاسوب للجمل التالية :

```
WRITELN(X,Y,Z);
WRITELN;
WRITELN(X,'**',Y,'**',Z);
WRITE(X:6:2);
WRITELN(Z:5:2);
WRITELN;
WRITELN('*****');;
```

12- اكتب برنامج بلغة باسكال لادخال ثلاثة أعداد وطباعتهم ، وطباعة العــــدد الاكبر من بين الأعداد الثلاث .

13- اكتب برنامج بلغة باسكال لايجاد جذور معادلة من الدرجة الثانية :

$AX^2 + BX + C = 0$

14- اكتب برنامج باسكال لحساب قيمة عمولة البائع المحصلة على مجمل مبيعاته S حسب القاعدة التالية :

$$\text{Commission} = \begin{cases} 0.02 \text{ OF S} & ; S<500 \\ 0.05 \text{ OF S} & ; 500<=S<=5000 \\ 0.10 \text{ OF S} & ; S>5000 \end{cases}$$

15- اكتب برنامج بلغة باسكال لحساب الاقتران التالي :

$$F(X) = X^2 + 2X^3 + X - 7$$

لجميع قيم X من 1 إلى 10 وبزيادة دورية مقدارها 0.25 .

16- تقدم 40 طالب لفحص في مادة المقدمة و سجلت العلامات ، والمطلوب كتابة برنامج باسكال لايجاد عدد الطلاب الناجحين وعدد الطلاب الراسبين وعدد الطلاب اللذين تزيد علاماتهم عن 80 .

17- اكتب برنامج لحساب الاقتران التالي :

$$F(X) = \begin{cases} 1 & ; X<0 \\ 4X^2+1 & ; 0<=X<=1 \\ 5X^3+3X+2 & ; X>1 \end{cases}$$

18- اكتب برنامج بلغة باسكال لطباعة وحساب مجموع الأعداد الزوجية من 2 إلى 50 .

19- اكتب برنامج بلغة باسكال لحساب كلا مما يلي :

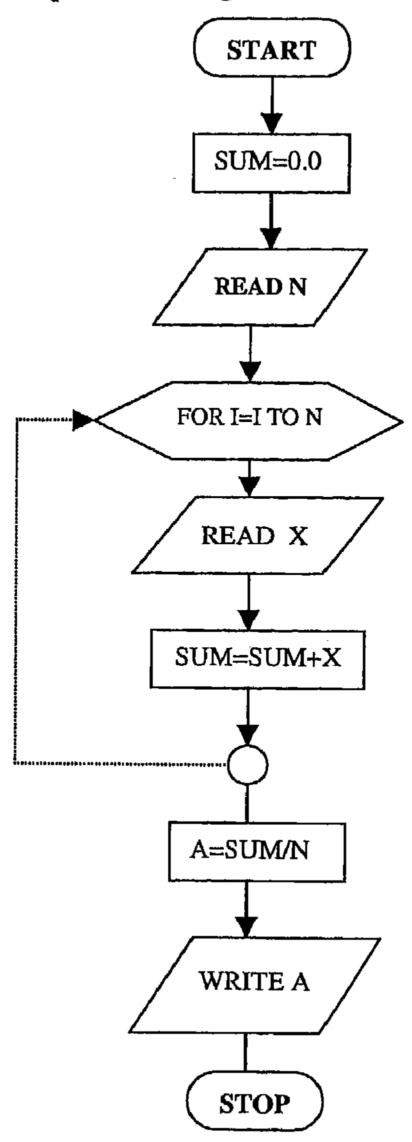
1) $SIN(X) = \frac{X}{1!} - \frac{X^3}{3!} + \frac{X^5}{5!} - \ldots.$

2) $\pi = 4(1-1/3 +1/5-1/7+1/9-\ldots..$

3) $e^x = 1 + X/1! + X^2/2! + X^3/3! + \ldots + X^3/N!$

20- اكتب برنامج بلغة باسكال لايجاد اكبر عنصر واصغر عنصر من عناصر مصفوفة A .

21- اكتب البرنامج بلغة باسكال المكافئ للمخطط التالي :



22- اكتب برنامج لايجاد اكبر عنصر في كل سطر من عناصر مصفوفة ما

23- اكتب برنامج لايجاد اصغر عنصر في كل عامود من اعمدة مصفوفة ما

24- اكتب برنامج كل جملة معادلات باستخدام طريقة غوص-جوردان

25- اكتب برنامج ترتيب مجموعة من القيم المدخلة بشكل عشوائي ترتيبا تصاعديا .

26-اكتب برنامج لحساب معدل علامات الطلاب المتقدمين لامتحان المقدمة ومقدار انحراف كل علامة عن المعدل : مع العلم ان

المعدل = مجموع علامات الطلبة / عدد الطلبة

انحراف كل علامة عن المعدل = علامة الطالب - المعدل

على ان تبدو شاشة المخرجات بالشكل التالي :

```
AVERAGE = .........
  العلامة              الانحراف
 SCORE            DEVIATION
.........          ................
.........          ...............

        *************
```

الفصل العاشر

البرمجة بلغة باسكال

PASCAL PROGRAMMING

بيانات المستخدم التعريفية — المصفوفات — البرامج الفرعية

السجلات والملفات

(10-1) بناء أنواع بيانات جديدة في لغة باسكال

تطرقنا في ما مضى الى الأنواع الأساسية من البيانات القياسية مثل :

Boolean data , char data , Real data , Integer data

ولكن هناك إضافة إلى ذلك أنواع من البيانات التي يمكن تسميتها بيانات المستخدم التعريفية User – defined – data وهذه المعطيات (البيانات) تكتب بطرق خاصة بحيث يمكن تحديد القيم التي يمكن أن تأخذها هذه الأنواع ثم يتم تعريف اسما جماعيا لهذا النوع ثم الإعلان عن متغيرات هذا النوع بحيث تأخذ هذه المتغيرات جزء أو كل من القيم التي سبق تحديدها وهذه الأنواع يمكن تقسيمها الى قسمين رئيسيين هما :

أ) البيانات معدودة القيم (التعددية)

ب) البيانات ذات المجال الجزئي (الجزئية)

(10-1-1) البيانات معدودة القيم Enumerated-Type data

هذا النوع من البيانات يتكون كما أوضحنا من مفردات من نوع واحد وتحمل اسما جماعيا يدل عليها وصورتها العامة هي :

```
TYPE name (data1, data2, ... , data n ) ;
```

حيث Name يمثل اسم نوع البيانات الجديد .

data 1 ⎫
.... ⎬ تمثل قائمة القيم التي يأخذها هذا النوع الجديد من البيانات .
data n ⎭

مثال :

```
TYPE  Months=( January, February, March, April, May,
June,  July,  August,  September, October, November,
December);
```

حيث نلاحظ في مثالنا هذا أن الشهر الممثل Months يمثل الاسم الجماعي الواصف لنوع البيانات TYPE وتكون هذه المفردات للبيانات مرتبة بشكل متسلسل وبالتالي يمكن مقارنتها ببعضها بإستخدام إشارات الأصغر والأكبر والعديد من إشارات المقارنة، كذلك يمكن أن نصرح عن المتغيرات من هذا النوع كما يلي:

```
VAR
      Y: Months ;
```

وبالتالي يمكن إسناد أي من القيم المعرفة في Months للمتغير Y أي يمكن ان نكتب في البرنامج ما يلي ; Y:=March، والمثال التالي يوضح استخدام إشارات العمليات المنطقية :

القيم المنطقية	التعبير المنطقي
True	March < April
True	November > = July
True	June < > July
True	SUCC (Junuary) = May
True	PRED (September) = Auguest

حيث استطعنا أن نستخدم بعض الاقترانات القياسية في لغة Pascal مثل SUCC و PRED مع البيانات القيم . وبما أن هذه البيانات مرتبة بشكل متسلسل فإنها تأخذ أرقام تصاعدية تبدأ بالصفر وتنتهي n-1 لأخر مفردة وكما في المثال التالي:

ord (Juanuary) = 0
ord (March) = 2
ord (July) = 6
ord (December) = 11

حيث تمثل المفردة الأولى الشهر الأول من السنة والثانية تمثل الشهر الثالث من السنة والمفردة الثالثة تمثل الشهر السابع من السنة والمفردة الأخيرة تمثل الشهر الثاني عشرة من السنة .

مع العلم انه في بعض الأحيان يمكن أن تستخدم في قيم الأنواع التعددية جملة من جمل التكرار For مثال :

```
FOR Months := Juanuary TO December DO ...,
```

أكتب برنامج بلغة باسكال لطباعة أيام الأسبوع باستخدام الأنواع (المعطيات) المعدودة القيم

```
PROGRAM DAYS (INPUT,OUTPUT);
TYPE Days_Of _Week=(Sat,Sun,Mon,Tus,Wen,Thur,Frid);
VAR
      Day:Days_Of_Week ;
      BEGIN
      FOR Day := Sat OF Frid DO
      WRITELN (day);
      END.
```

(2-1-10) البيانات الجزئية Subrange-Type-data

أعطيت اسم الجزئية كونها تمثل جزء من سلسلة متكاملة من مفردات المعطيات البسيطة مثل البيانات (الرمزية او المنطقية او الصحيحة) او البيانات المعدودة حيث القيم السابقة ومفردات هذا النوع من البيانات تكون مرتبة ترتيبا تصاعديا وصيغتها العامة هي كما يلي :

```
TYPE name=First data..last data ;
```

حيث تمثل First data المفردة الأولى من مجموعة البيانات الجزئية بينما تمثـل Last data المفردة الأخيرة على أن يفصل بينها نقطتان متجاورتان وبحيث تقـع بين المفردة الأولى والأخيرة بمجموعة البيانات الجزئية المطلوبة بما فيها المفردتان الأولى والأخيرة .

مثال (2) : لبيان العطلة الصيفية يمكن ان نعبر عنها بما يلي :

```
TYPE months=(January, February, March, April, May,
June, Julay, August, September, October, November,
December);
Holidays = June..Augest ;
```

حيث بينا في المثال أعلاه كيفية استعمال البيانات الجزئية تحت اسم العطلـة Holidays والتي تقع مفرداتها بين شهري تموز وآب والتي تمثل العطلة الصيفيـة السنوية .

ويمكن ان تستخدم أيضا البيانات الصحيحة كبيانات جزئية

```
Months = 1.....12 ;
```

أو يمكن كذلك أن تستخدم أيضا البيانات الرمزية CHAR كبيانات جزئية :

```
CLASS NAME = 'ALI' .. 'AHMAD' ;
```

وكما أوضحنا سابقا فإنه يمكن استخدام بعض جمل الشرط والتكرار مـع هـذه البيانات مثل جملة FOR وجملة CASE .

مثال (3) : استخدام جملة CASE مع المعطيات الجماعية التعددية المعرفة من قبل المستخدم تكون على الشكل التالي :

```
CASE Months OF
    February : WRITELN ('Start Semester');
    March    : WRITELN ('First Exam') ;
    April    : WRITELN (Second Exam') ;
    May      : WRITELN ('Final Exam');
    END;
```

(2-10) المصفوفات ARRAYS

(1-2-10) مفهوم المصفوفة في لغات البرمجة

عندما تستخدم متغيراً بسيطاً في أحد البرامج ، كما رأينا سابقاً ، يقوم الحاسوب بتحديد موقعاً تخزينياً واحداً يخزن فيه قيمة هذا المتغير ، فمثلاً إذا كان قيمة المتغير A تساوي 78 (عدد صحيح) ، عندئذ في كل مرة يظهر المتغير A أثناء تنفيذ البرنامج يقوم الحاسوب باستخدام القيمة المخزنة لـ A والتي هي 78 .

اذا كان A متغير يعبر عن درجات 20 طالب في مادة ما ، فإنه في مثل هذه الحالة نحتاج الى 20 موقعا تخزينيا وكل موقع يحتاج الى اسم متغير مختلف ، لتمييز القيم التخزينية عن بعضها البعض ، وللتغلب على مشكلة استخدام اسماء متغيرات مختلفة لتسمية المواقع التخزينية ، يستخدم اسم مشترك لكل المواقع ويشار الى كل موقع برقم الذي يدعى دليل او مؤشر Index يدل على موقع العنصر ، وهذا يقابل في الرياضيات مصفوفة احادية البعد ويرمز لها بـ A_i حيث $i=1,2,...,20$

ويمكن تعميم ما سبق على المتغيرات لبيانات متشابهة في النوع والمختلفة في القيم لأكثر من صفة ، فمثلا نريد التعبير عن درجات 20 طالب في مادة ما لثلاثة شعب، عندئذ نرمز بـ A_{ij} حيث $i=1,2,3$ رقم الشعب و $j=1,2,...,20$ رقم الطالب، وهذا ما يقابل في الرياضيات مصفوفة ثنائية البعد ، وبالتالي عندما نكتب $A_{24}=70$ نقصد بها ان درجة الطالب الرابع من الشعبة الثانية تعادل 70 ، ويمكن تشبيه عملية التخزين بالمصفوفة التالية :

-	1	2	3	4	...	20
1	80	30	50	33	...	60
2	83	45	80	70	...	55
3	72	90	74	65	...	84

والغاية مما سبق هو التعامل مع البيانات المتشابهة بالنوع من حيث نوعية البيانات (حقيقية،صحيحة،رمزية) ومن حيث ما يقصد بها واقعيا مثل (درجات عدد من

الطلبة في شعب مختلفة وفي جامعات مختلفة) ، والمختلفة بالقيم بشكل سهل ويسر ويعطي البرنامج مظهرا بسيطا غير معقدا .

(10-2-2) تمثيل عناصر المصفوفة في لغة باسكال

1) المصفوفة ذات البعد الواحد وتأخذ الصيغة التالية :

```
Array Name  [Index ] ;
```

حيث ARRAY Name اسم المتغير والذي يجب التصريح عنه في بداية البرنامج.

Index دليل والذي يمكن ان يكون ثابت عددي او متغير عددي او تعبير حسابي .

والدليل يجب ان لا تتجاوز قيمته الحد الأقصى المحدد في صيغة الاعلان

مثال (4) :

```
1) GRADE [ 1 ] := 50 ;
2) GRADE [ 2 ] := 70 ;
3) SUM:=SUM+A[ i ] ;
4) READ (A[ i ] );
```

2) المصفوفة المتعددة الأبعاد وتأخذ الصيغة التالية :

```
Array Name [ Index1, Index2 ·····] ;
```

حيث Array Name اسم المتغير والذي يجب التصريح عنه في بداية البرنامج .

Index 1
Index 2
:
:
:

حيث تمثل أدلة والتي يمكن أن تكون ثابت عددي أو متغير عددي أو تعبير حسابي ، ويجب أن لا تتجاوز هذه الأدلة الحد الأقصى المحدد في صيغة الإعلان .

مثال (5) :

```
1) GRADE [ 1,3 ] := 70 ;
2) X [ 5,20,4 ] := 80 ;
3) SUM = SUM + [ i,j ] ;
4) WRITELN (A [ i,j ]) ;
```

(3-2-10) الاعلان (التصريح) عن المصفوفات .

يتم الاعلان عن المصفوفات بنفس طريقة الإعلان عن المتغيرات مع إضافة الإعلان عن أبعاد المصفوفة .

أ) الصيغة العامة للإعلان عن مصفوفة أحادية البعد

```
VAR
     Array Name : ARRAY [ N1…N2] OF Type ;
```

حيث : Array Name اسم المصفوفة .

N1…N2	مدى العناصر في المصفوفة أي ان N1 تعبر عن رقم العنصر الاول من المصفوفة ، N2 العنصر الاخير من المصفوفة .
Type	نوع عناصر المصفوفات

مثال (6) :

```
VAR
     GRADE : ARRAY [1..20] OF INTEGER ;
```

نلاحظ ان عناصر المصفوفة من النوع الصحيح وان مدى الدليل ستكون من 1 وحتى 20 .

ب) الصيغة العامة للإعلان عن مصفوفة ثنائية البعد :

```
VAR
     Array Name : [n1..n2,m1..m2] OF Type ;
```

حيث Array Name اسم المصفوفة

n1..n2	مدى العناصر للبعد الأول (عناصر الصفوف في المصفوفة)
m1..m2	مدى العناصر للبعد الثاني (عناصر أعمدة المصفوفة)

مثال (7) :

```
VAR
     Grade : Array [ 1..3,1..20] OF REAL ;
```

نلاحظ أن البعد الأول هو من 1 وحتى 3 أي ثلاثة صفوف و البعد الثاني هو من 1 وحتى 20 أي عشرون عمودا.

ج) الصيغة العامة للإعلان عن مصفوفة متعددة الأبعاد :

```
VAR
     Array Name : [n1..n2,m1..m2,L1..L2,…] OF Type ;
```

(10–2–4) أمثلة على استخدام ومعالجة المصفوفات :

مثال (8) : البرنامج التالي لحساب معدل درجات الطلبة في مادة المقدمة

لنرمز بـ A_i لعلامة الطالب i

N لعدد الطلبة

AVERAGE لتجميع علامات الطالب

i دليل كعداد

```
PROGRAM EXAMP (INPUT,OUTPUT) ;
VAR
     A : ARRAY [ 1..50] OF INTEGER ;
     N,I : INTEGER ;
     SUM, AVERAGE : REAL ;
BEGIN
     WRITE ('Enter Number Students :') ;
     READLN (N) ;
     FOR I:=1 TO N  DO BEGIN
          WRITE ('A[', I ,]=');
          READLN (A[I]) ;
          END;
     SUM:=0.0 ;
     FOR I :=1 TO N DO
```

```
        SUM:=SUM+A[I] ;
    AVERAGE:=SUM/N ;
    WRITELN ('AVERAGE=' , AVERAGE:5:2) ;
END.
```

وعند التنفيذ ستبدو شاشة المخرجات كما يلي :

```
Enter Number Student :5
A[1]=65
A[2]=73
A[3]=55
A[4]=80
A[5]=97
AVERAGE=74.00
```

مثال (9) : اكتب برنامج بلغة باسكال لإدخال خمسة أعداد ثم ترتيبهم تنازليا وطباعة العناصر المدخلة ، والعناصر المرتبة .

```
PROGRAM EXAM70 (INPUT,OUTPUT) ;
VAR
    X : ARRAY [1..50] OF INTEGER ;
    I,N,T : INTEGER ;
BEGIN
    WRITE ('Enter the Value of N=') ;
    READLN (N) ;
    FOR I :=1 TO N DO
        BEGIN
            WRITE ('X[',I,']=') ;
            READLN (X[I]) ;
        END;
    WRITELN ('Unsorted Array is : ') ;
    FOR I := 1 TO N DO
        WRITE (X[I],'  ') ;
    FOR I :=1 TO N-1 DO
        BEGIN
            FOR K:= I+1 TO N DO
            BEGIN
```

```
                IF X[I]<=X[K] THEN
                BEGIN
                    T:=X[I] ;
                    X[I]:=X[K] ;
                    X[K]:=T ;
                END ;
            END;
        END;
    WRITELN ('Sorted Array is :') ;
    FOR I:=1 TO N DO WRITE (X[I],'  ') ;
END.
```

وعند التنفيذ ستبدو شاشة المخرجات كما يلي :

```
Enter the Value of N=5
58
70
34
55
91
Unsorted Array is :
58    70    34    55    91
Sorted Array is :
91    70    58    55    34
```

مثال (10) اكتب برنامج لإدخال عناصر مصفوفة تتكون من خمسة صفوف وستة أعمدة ثم حساب مجموع عناصر كل صف .

```
PROGRAM EXAMEL (INPUT,OUTPUT) ;
CONST
    ROWSIZE = 5 ; COLSIZE=6 ;
VAR
    X: ARRAY [1..Rowsize,1..Colsize] OF INTEGER ;
    SUMROW : ARRAY [1..ROWSIZE] ;
    I,J : INTEGER ;
BEGIN
    FOR I:=1 TO 5 DO
    BEGIN
```

```
            FOR J:= 1 TO 6 DO
            READ (X[I,J]) ;
            WRITELN ;
      END;
      FOR I := 1 TO 5 DO
      BEGIN
            SUMROW [I] := 0.0
            FOR J := 1 TO 6 DO
                  SUMROW [I] = SUMROW [I] + X[I,J] ;
      END;
      WRITELN ('SUM OF EACH ROW :') ;
      FOR I := 1 TO 5 DO
      WRITELN ('SUM OF ROW (',I,') = ',SUMROW [I]);
END.
```

مثال (11) : اكتب برنامج بلغة باسكال لقراءة عناصر مصفوفة A ثم جد حاصل ضرب هذه المصفوفة بثابت عددي S علما بأن عدد صفوفها 7 وعدد أعمدتها هو 6 ثم بعد ذلك خزن عناصر المصفوفة الجديدة الناتجة من عملية الضرب في مصفوفة جديدة B

```
PROGRAM EXAMPEL (INPUT,OUTPUT) ;
VAR  A : ARRAY [1..7,1..6] OF REAL ;
      B : ARRAY [1..7,1..6] OF REAL ;
      N : M  :   INTEGER ;
      S : REAL ;
BEGIN
      READLN (S) ;
      FOR N := 1 TO 7 DO
            FOR M := 1 TO 6 DO
            BEGIN
                  READLN A[N,M] ;
                  B [N,M] := S* A[N,M]
            END;
FOR N:= 1 TO 7 DO
      BEGIN
```

```
            FOR M:= 1  TO 6 DO
            WRITE (B [N,M] ) ;
            WRITELN ;
        END;
END .
```

(3-10) البرامج الفرعية SubPrograms

إن استخدام أسلوب التصميم الهرمي للبرمجة هو الأسلوب الناجح والكفء في لغة باسكال حيث يوفر هذا الأسلوب سهولة في التصميم وسهولة في كتابة البرامج مع إمكانية التأكد من صحة البرنامج الذي تترابط أجزائه بصورة منطقية صحيحة حيث يتم بإستخدام أسلوب TOP-DOWN والذي يشير إلى تقسيم البرنامج إلى أجزاء رئيسية بحيث يؤدي كل جزء وظيفة معينة في البرنامج ، وبربط هذه الأجزاء فيما بينها يمكن الوصول إلى برنامج متكامل ، وكذلك يمكن الرجوع إلى أي جزء من البرنامج في أي وقت . هذه الأجزاء من البرنامج المترابطة تسمى في لغة البرمجة (البرامج الفرعية) (Sub program) ، وفي لغة Pascal تسمى هذه البرامج الفرعية بـ (Procedure) الإجراء أو (Function) الاقتران.

هذه البرامج الفرعية تكون سهلة التصميم حيث يقوم المبرمج بإعدادها بشكل منفصل ثم يتم تجميعها وربطها بالبرنامج العام ، ولكون هذه البرامج الفرعية صغيرة ومستقلة وغير متشعبه فإنها تمكن المبرمج من التأكد من صحتها وصحة نتائجها بسهولة وللبرامج الفرعية فائدة كبيرة وهامة في المسائل المعقدة خاصة والتي تحتوي على التكرار المتعدد لبعض جمل باسكال ، حيث يتم التعويض عن التكرار المتعدد بأسلوب استدعاء الاجراء او الاقترانات عند الحاجة وبهذا يتم اختصار حجم البرنامج الى حد كبير . وسوف نقوم بشرح وتفصيل كل من الإجراءات والاقترانات ومعرفة استخداماتها والفرق فيما بينها .

(1-3-10) الإجراءات Procedures

في لغة باسكال Pascal هناك نوعان من الإجراءات :

الأول : هو الإجراء الداخلي Internal وهو الذي يتم إعداده داخل البرنامج الرئيسي ويكون موقعه بعد تعريف جميع الاقترانات في البرنامج .

الثاني : هو الإجراء الخارجي External وهو الذي يتم إعداده خارج البرنامج الرئيسي وبشكل مستقل ومنفصل تماما ويتم استخدامه بوضع اسمه مع المتغيرات المستخدمة في تعريف الإجراء داخل البرنامج الرئيسي .

إن الإجراءات تستخدم عند الحاجة لحساب قيمة لأكثر من عنصر واحد عند استدعاء البرنامج الفرعي لاستخدامها في البرنامج الرئيسي حيث يبدأ تعريف الإجراء باستخدام كلمة Procedure ثم يليها اسم الإجراء وبعدها يتم وضع قائمة المتغيرات التي يعتمد عليها الإجراء وبحيث يكون ترتيب وتنظيم الإجراء مثل تنظيم البرنامج الاعتيادي والصيغة العامة للإجراء Procedure هو كما يلي :

```
PROCEDURE name ( Parameter List );
Local Variable declaration section
BEGIN
------------  }
------------  }   جمل باسكال
------------  }   تمثل جسم الإجراء
END.
```

حيث :

PROCEDURE عنوان الإجراء وهي كلمة محجوزة

name اسم الإجراء الذي يتبع الكلمة المحجوزة PROCEDURE

Parameter List — تمثل قائمة المتغيرات أو باراميترات الإجراء والتي يتم تعويضـــها في جملة الاستدعاء في البرنامج الرئيسي بالمتغيرات الفعلية الـــتي يجب ان تتطابق مع المتغيرات المحايدة من حيث النوع والعدد .

Local Variable declaration section — مقطع الاعلان عن المتغيرات التي ستستخدم داخـــل الاجـــراء فقط وتدعى متغيرات موضعية ، بينما المتغيرات التي يصـــــرح عنها في البرنامج الرئيسي فتدعى المتغيرات الشاملة والتي يكـــن استخدامها في مكان من البرنامج الرئيسي والبرامج الفرعية .

و يجب أن يتبع كل بارميتر في Parameter List نوعه ، وتقسم عـــــادة هـــذه البارامترات الى نوعين :

- متغيرات مدخلات : ويتم تعريف قيمتها من خلال المتغيرات الفعليـــــة الـــواردة في جملة الاستدعاء في البرنامج الرئيسي والتي تقابلها في الترتيب وتتطـــــابق معـــها في النوع .
- متغيرات مخرجات : ويتم حسابها داخل الإجراء وتســـند قيمتــــها إلى المتغـــير الفعلي في جملة الاستدعاء في البرنامج الرئيسي .
- وعادة نرتب قائمة متغيرات الإجراء المدخلات أولا ومن ثم الكلمة المحجـــوزة VAR وبعدها متغيرات المخرجات .

مثال (12) أكتب برنامج بلغة باسكال لاختيار العدد الأكبر من بين ثلاثة أعـــداد معلومة مستخدما PROCEDURE

```
PROCEDURE MAXIMUM (A,B,C : INTEGER) ;
VAR
      MAX : INTEGER ;
BEGIN
      IF A>B THEN MAX:=A ELSE MAX:=B ;
      IF C>MAX THEN MAX:=C ;
      WRITELN('THE MAXIMUM =' ,MAX)
END;
```

مثال (13) اكتب برنامج بلغة باسكال (إجرائي) لتبديل قيم متغيرين فيما بينهما.

نلاحظ في هذا السؤال انه يجب علينا أن نستخدم متغير وسيط بين المتغـــيرين أعلاه لكي نقوم بتحويل قيم أحدهما للآخر ولهذا وجب عند استخدام متغير وسيط تعريفه وتحديد نوعه في مقطع التصريح داخل برنامج الإجراء

```
PROCEDURE SWAP (X,Y : REAL) ;
VAR
      TEMP : REAL ;
BEGIN
      TEMP :=X; X:=Y; Y:=TEMP ;
END;
```

مثال (14) اكتب برنامج بلغة باسكال لإيجاد اكبر عدد وأصغر عدد من مجموعة أعداد باستخدام الإجراء .

```
PROGRAM F_MAX_MIN(INPUT,OUTPUT) ;
VAR
      N      :INTEGER ;
PROCEDURE MAX_MIN (NUM  : INTEGER);
      MAX,MIN,X  : REAL ;
      I               : INTEGER ;
BEGIN
      WRITELN ('ENTER',NUM,'NUMBER PLEASE ..');
      READ (X) ;
      MAX :=X ;
      MIN  := X ;
      FOR I :=  1 TO NUM-1 DO
      BEGIN
            READ (X) ;
            IF X >MAX
                  THEN
                        MAX :=X ;
            IF (X<MIN)
                  THEN
```

```
                    MIN := X ;
      END;
    WRITELN('MAX IS:',MAX :4:2,',MIN IS:',MIN:4:2);
END;
BEGIN
      CLRSCR ;
      WRITELN('PLEASE ENTER HOW NUMBER YOU WANT:');
      READ (N) ;
      MAX MIN (N) ;
      REPEAT UNTIL REYPAESSED ;
END.
```

مثال (15) أكتب برنامج بلغة باسكال لحساب مربع عدد حقيقي X لجميع القيم المحصورة بين 0 و 1 بزيادة مقدارها 0.1 باستخدام الاجراء .

```
PROGRAM EXAMPEL (INPUT,OUTPUT) ;
 VAR
      X,Y  : REAL ;
PROCEDURE Sqrnumber (X : REAL ; VAR X2 :REAL);
      BEGIN
            X2 := X * X ;
      END;
BEGIN    (* MAIN PROGRAM *)
      X:=0.0 ;
      WHILE X < 1.0 DO BEGIN
      Sqrnumber (X,Y);
      WRITELN ('SQR(',X,')=',Y:S:2) ;
      X:=X+0.1 ; END;
END.
```

(2-3-10) الاقترانات FUNCTIONS

إن الاقتران في لغة باسكال يستخدم عادة لحساب قيمة معينة واحدة بينما يتم استخدام الإجراء لحساب مجموعة من القيم ، حيث يتم تعريف الاقتران عادة في بداية البرنامج بعد تعريف الجزء الخاص بوحدات التخزين اللازمة للبرنامج، على ان يعامل اسم الاقتران كاسم متغير في البرنامج ، ولهذا فإن تنفيذ

الاقتران في البرنامج يتم فورا بمجرد ذكر اسم الاقتران ، لهذا فمن الواجب ذكر نوع الاقتران هل هو حقيقي ، ام صحيح ، ام رمزي ، ام منطقي .

علما ان هناك نوعان من الاقترانات وهما :

1. الاقترانات المكتبية Library Functions

2. اقترانات المبرمج التعريفية Internal defined functions

و الآن سوف نقوم بشرح هذين النوعين من الاقترانات :

(10-3-2-1) الاقترانات المكتبية Library Functions

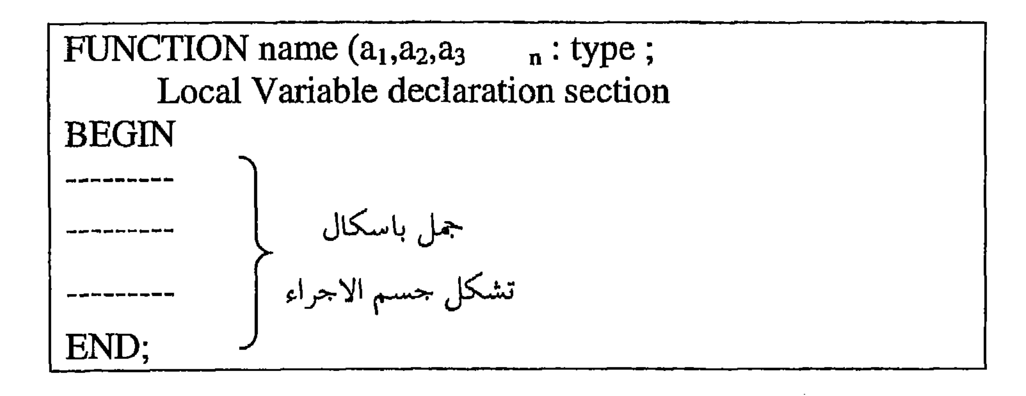
هذه الاقترانات تكون معرفة في لغة باسكال أصلا وهي مجموعة من الاقترانات القياسية الجاهزة للاستخدام بمجرد ذكر اسمها وفي أي موقع من مواقع البرنامج ولا يوجد هناك ضرورة لتعريفهما في البرنامج من هذه الاقترانات على سبيل المثال :

SQRT (X)	SQR (X)	ABS (X)
LN (X)	COS (X)	SIW (X)
ROUND (X)	TRURC (X)	EXP (X)
CHR (X)	ORD (X)	ARCTAN (X)
ODD (X)	SUCC (X)	PRED (X)

من الضروري في هذه الاقترانات مراعاة ان تكون المتغيرات الفعلية المستخدمة مطابقة فعلا لنوع المتغيرات المحايدة المعرفة لكل اقتران قياسي وقد تناولنا هذه الاقتراتات في الفقرة (9-2-8) .

(10-3-2-1) اقترانات المبرمج التعريفي Internal defined Functions

هذه الاقترانات يتم كتابتها وتعريفها داخل البرنامج وفي بدايته ومن ثم استخدامه داخل الجزء التنفيذي من البرنامج ، ولهذا يسمى هذا النوع من الاقترانات ايضا بالاقترانات الداخلية وصيغته العامة :



```
FUNCTION name (a1,a2,a3      n : type ;
      Local Variable declaration section
BEGIN
---------  ⎫
---------  ⎬  جمل باسكال
---------  ⎪  تشكل جسم الاجراء
END;       ⎭
```

مثال (16) : اكتب برنامج اقتران لإيجاد معدل قيمتين.

```
FUNCTION AVG (x,y :REAL ): REAL ;
BEGIN
      AVR:= (X + Y) / 2.0 ;
END;
```

مثال (17) : اكتب برنامج اقتران لإيجاد مساحة دائرة معلوم نصف قطرها .

```
FUNCTION AREA (Y:REAL):REAL ;
BEGIN
      AREA :=3.14159*Y*Y ;
END;
```

مثال (18) : أكتب برنامج اقتران لحساب القيمة العظمى لقيمتين معلومتين

```
FUNCTION LARGER (D,E : REAL) : REAL ;
BEGIN
      IF D>E THEN
      LARGER :=D
      ELSE
      LARGER :=E ;
END;
```

مثال (19) اكتب برنامج اقتران لحساب القيمة المطلقة | S | و المعرفة على النحو التالي :

$$Y(S) = \begin{cases} S & \text{if} \quad S > 0 \\ -S & \text{if} \quad S < 0 \\ 0 & \text{if} \quad S = 0 \end{cases}$$

```
FUNCTION
      ABSLT (S:REAL):REAL ;
BEGIN
      IF S<0 THEN ABSLT:=-S ELSE
      IF S=0 THEN ABSLT:=O ELSE
      ABSLT:=S ;
END;
```

مثال (**20**) : اكتب برنامج بلغة باسكال لحساب معدل درجات 30 طالب باستخدام الإجراء لحساب المعدل .

```
PROGRAM EXAM17 (INPUT,OUTPUT) ;
TYPE
      Vector =ARRAY[1..30]OF REAL ;
VAR
      GRADE :Vector ;
      AVG    : REAL ;
      I      : INTEGER ;
PROCEDURE Avereg (N:Integer; Y:Vector; VAR
Z:REAL);
VAR
      SUM : REAL ;
BEGIN
      SUM :=0.0 ;
      FOR I:=1 TO N  DO
      SUM:=SUM + Y[I] ;
      Z:=SUM/N ;
END;
BEGIN
      FOR I:=1 TO 30 DO READ (GRADE[I]);
      Avereg (5, GRADE , AVG);
      WRITELN (AVG:4:2);
END.
```

مثال (21) اكتب برنامج لحساب مضروب عدد باستخدام الاقتران حيث :

$$N! = \begin{cases} 1 & ; n=0 \\ 1*2*3*....*n & ; n \quad 0 \end{cases}$$

```
PROGRAM FACTRIAL (INPUT,OUTPUT);
VAR
      N  : INTEGER ;
FUNCTION FAC (K : INTEGER) : INTEGER ;
VAR
      F,I : INTEGER ;
BEGIN
      F :=1 ;
      FOR I := 1 TO K DO
      F := F*I ;
      FAC := F ;
END ;
BEGIN
      CLRSCR ;
      WRITE ('N=') ; READLN (N) ;
      WRITE (N,'!=', FAC(N)) ;
      READLN ;
END.
```

مثال (22) اكتب برنامج بلغة باسكال لطباعة قيم كل من X ، X^3 لجميع قيـــم X من 1 الى 5 بزيادة مقدارها 0.5 مستخدما الاقتران لحساب X^3

```
PROGRAM EXAMPEL (INPUT,OUTPUT) ;
VAR
      X,Y : REAL ;
FUNCTION CUBE (A:REAL) : REAL ;
      BEGIN
            CUBE:=A*A*A ;
      END;
BEGIN (* MAIN PROGRAM *)
```

```
X:= 1.0 ;
WRITELN ('X            Y');
WRITELN ('---          ---');
WHILE X <=5.0 DO
    BEGIN
        Y:=CUBE (X)
        WRITELN (X:5:2 , '     ' , Y:6:2);
    END;
END.
```

(4-10) السجلات RECORDS

ان السجلات هي من المعطيات البنائية (Structured Types) في لغة باسكال وتعتبر ذات مقدرة عالية في معالجة البيانات .

لقد تطرقنا فيما سبق الى معالجة معلومات من نفس النوع (رقمية) أو حرفية أو رمزية أو مصفوفات ولكن في بعض الأحيان نحتاج لمعالجة معلومات مختلفة النوع، ففي المصفوفات مثلاً كنا نعالج منظومات مرتبطة فيما بينها مثل درجات الطلاب وأسماء الطلاب وكانت الطريقة معقدة في بعض الأحيان ، لذلك فمن الأفضل التعامل مع السجلات التي تمكننا من معالجة معلومات من أنواع مختلفة ، حيث أن المعلومات المختلفة في السجل تحتل مجالات خاصة بكل نوع تدعى حقول فمثلاً في حالة مجموعة من الطلبة وأرقامهم وموادهم وامتحاناتهم نجد أن اسم الطالب يمثل حقل للأسماء ورقم الطالب يمثل حقل للأرقام ومادة الطالب تمثل حقل للمواد ودرجات الطالب تمثل حقل للمعلومات وهكذا .

فلو فرضنا أن السجل الذي يحوي أسماء وأرقام ودرجات ومواد الطلبة يكون بالشكل التالي :

اسم الطالب name	رقم الطالب No.	المادة Course	الامتحان الاول First Ex	الامتحان الثاني Second Ex	الامتحان النهائي Final Ex	المجموع sum

فهذا الجدول يمثل حقول ولكل حقل خصوصية خاصة في السجل ولهذا نستطيع

تعريف السجل من خلال الجدول أعلاه بإحدى الطريقتين :

الطريقة الأولى : يتم تعريف السجل وإدخاله كجزء من الإعلان عن المتغيرات (Variables) وهذا التعريف يتم الاستفادة منه في تعريف أنواع متغيرة من السجلات المستقلة تماما كما يتم تعريف المتغيرات البسيطة والمنظومات المتغيرة ولهذا فإن الصورة العامة لتعريف السجل في هذه الحالة يكون على الشكل التالي :

```
VAR name : RECORD
            field 1 ;
            field 2 ;
            field 3 ;
                :
            field n ;
END.
```

وفي مثالنا السابق فان field 1 يمثل الاسم

field 2 يمثل الرقم

field 3 يمثل المادة

field 4 يمثل العلامة

field 5 يمثل المجموع

وهذه جميعها تمثل الإعلان عن المجالات المختلفة الداخلة في السجل

و name تمثل اسم السجل

علما بأن المجالات المختلفة الداخلية في السجل تكتب بطريقة تماثل طريقة الإعلان عن المتغيرات المستقلة أي :

```
Field name : type ;
```

حيث Filed Name اسم المجال .

TYPE نوع المجال .

مع ملاحظة وجود الفارزة المنقوطة التي تفصل بين المجالات المستقلة ضمن السجل الواحد

مثال (23) لدينا سجل طالب يتكون من رقم الطالب ورمز معدلـــه الـتراكمي ودرجته في مادة معينة ، يمكن تمثيل ذلك في متغير سجلي يسمى Student وعلى أساس أن الرقم يتألف من (أرقام صحيحة) والمعدل التراكمي يتألف من (حـروف) ودرجة الطالب تتألف من (أعداد عشرية) ولهذا يمكن كتابة السجل كما يلي :

```
VAR
     STUDENT          : RECORD
     STUDNO           : INTEGER ;
     POINT            : CHAR ;
     TOTGRADE         : REAL ;
     END ;
```

نلاحظ ان هذا السجل يحوي على ثـــلاث مجـــالات المجـــال الأول سمـــي STUDNO رقم الطالب ، المجال الثاني سمي POINT والذي احتـــوى رمـــز معدل الطالب التراكمي في المادة فمثلا بالحروف CHAR والمجال الثـــالث سمـــي TOTGRADE والذي يحوي درجة الطالب في المقرر المعين ومن الضـــروري ان تكون مكونات السجل من الأنواع المعروفة والتي يتم تعريفها في البرنـــامج وقبـــل تعريف السجل .

الطريقة الثانية : يتم تعر يف السجل على أساس النوع (TYPE) وليس كمتغـــير ثم بعدها يتم الإعلان عن المتغيرات السجلية وهذا التعريف متطابق مـــع تعريـــف المصفوفات (Arrays) ، ولهذا فهي اكثر انتشارا من الطريقة الأولى حيث إننـــا نستطيع تعريف اكثر من متغير سجلي من النوع الـــذي تم تعريفـــه وهنـــا يتـــم الاستعاضة عن كلمة VAR بكلمة TYPE والصيغة العامة لهذه الحالة :

```
TYPE
     name =    RECORD
                  field 1;
                  field 2;
                     :
                  field n;
     END
```

حيث NAME اسم السجل

field 1
field 2
......
field n

تمثل الإعلان عن المجالات المختلفة الداخلية في السجل

مثال (24) : لدينا سجل طالب يتكون من رقم الطالب ورمز معدلـه الـتراكمي ودرجته في مادة معينة ، يمكن تمثيل ذلك في متغير سجلي يسـمى STUDENT وعلى أساس أن الرقم يتألف من (أرقام صحيحة)، والمعدل التراكمي يتألف مـن حروف، ودرجة الطالب تتألف من أعداد عشرية، لذلك يتم تعريف نوع سـجلي أسمه GRADE ومتغير سجلي أسمه STUDENT يكون من نـوع GRADE وكما يلي :

```
TYPE      GRADE : RECORD
          STUDENT : INTEGER ;
          POINT : CHAR ;
          TOTGRADE : REAL ;
          END;
VAR       STUDENT : GRADE ;
```

(5-10) الملفات Files

الملف هو عبارة عن مجموعة من السجلات التي تضم عدد مـن الحقـول المختلفة النوع والطول ، ينتهي كل ملف في لغة باسكال بعلاقة نهاية الملـف EOF أي (End Of File) وتتصف جميع الملفات بلغة باسكال بأنها ملفات تتابعيـة أي ان النفاذ الى الملف يتم بالتسلسل أي عنصر تلو الاخر .

ويتم الإعلان (التصريح) عن الملف بالصيغة العامة التالية :

```
TYPE    file name = FILE OF Type ;
```

حيث file name اسم الملف

Type نوع الملف والذي يمكـن ان تكـون , CHAR , REAL , INTEGER , TEXT

ويمكن التمييز بين نوعين من الملفات :

1. ملفات الإدخال Input Files : وهي ملفات معدة للقراءة من قبل البرنـــامج الرئيسي .

2. ملفات الإخراج Output Files : وهي ملفات معـــدة لتخزيـــن وحفـــظ البيانات حيث يتم تسجيل البيانات عليها ليتم استدعاءها في الـــبرامج وقـــت الحاجة .

وللتعامل مع الملفات ومعالجتها يستخدم جمل مختلفة أهمها :

1) REWRITE (F) لفتح أو إنشاء ملف جديد فارغ باسم F .

2) WRITE (F,X) لكتابة القيمة X على الملف F .

3) ASSIGN(F,'file name') تنسيب اسم الملف F إلى ملـــف خـــارجي باسم file name .

4) CLOS (F) إغلاق ملف مفتوح .

5) RENAME يستخدم لتغيير اسم ملف .

6) RESET (F) يفتح ملف موجود مسبقاً وذلك للقراءة

7) READ (F,X) لقراءة القيمة X من الملف F .

8) EOF إشارة إلى نهاية الملف .

9) EOLN إشارة إلى نهاية السطر .

بالإضافة إلى جمل أخرى .

مثال (25) : البرنامج التالي لإنشاء ملف يحفظ نتائج البرنامج الذي يحسب مربـــع عدد

```
PROGRAME EXAMPEL1(INPUT,OUTPUT,FILES) ;
TYPE SQRFILE = FILE OF INTEGER ;
VAR
      X: SQRFILE ;
      I,J,N : INTEGER ;
BEGIN
      CLRSCR ;
```

```
        WRITELN ('ENTER N = ' ) ;
        READLN (N) ;
        ASSIGN (X,'A: TEST . DAT ') ;
        REWRITE (X) ;
        I:=1 ;
        WHILE I <= N DO
              BEGIN
                    J:=SQR(I) ;
                    WRITELN (I,'  ',J) ;
                    WRITE (X,J) ;
                    I :=I+1 ;
              END;
        CLOSE (X) ;
END.
```

من المثال السابق نلاحظ أننا صرحنا عن الملف على انه من نوع الصحيح في بداية البرنامج وباستخدام الجملة ASSIGN اسندنا الملف X إلى الملف TEST.DAT على القرص المرن الموجود على السواقة A ، ثم ضمن حلقة التكرار المشروط WHILE حسبنا مربعات الأعداد وضمن جملة WRITE (X,J) خزنا هذه المربعات J على الملف X ثم أغلقنا الملف المفتوح باستخدام الجملة CLOSE .

مثال (26) : البرنامج التالي لقراءة الملف الذي أنشأناه في المثال السابق ثم جمع تلك القيم وعرضها على الشاشة :

```
PROGRAM  EXAMPEL2(INPUT,OUTPUT,FILES) ;
TYPE SUMFILE = FILE OF INTEGER ;
VAR
        X: SUMFILE  ;
        I,SUM : INTEGER ;
BEGIN
        CLRSCR ;
        ASSIGN (X,'A: TEST , DAT ') ;
        RESET (X) ;
```

```
        SUM=0 ;
        WHIL NOT EOF (X) DO
            BEGIN
                READ (X,I) ;
                WRITELN ('I=',I) ;
                SUM :=SUM+I ;
            END;
        CLOSE (X) ;
        WRITELN ('THE TOTAL VALUES = ',SUM:6) ;
END.
```

من المثال السابق نلاحظ أننا باستخدام الجملة RESET فتحنا الملف X المخزن مسبقاً ثم باستخدام الجملة READ(X,I) قرأنا القيمة I من الملف X ثم جمعنا هذه القيم (التي تمثل مربعات الأعداد) أظهرنا النتائج على الشاشة بعد أن أغلقنا الملف المفتوح باستخدام الجملة CLOSE (X) .

مثال (27) : البرنامج التالي يتضمن إنشاء ملف لإدخال البيانات على شكل سجلات يتألف هذا السجل من رقم الزبون ، اسم الزبون ، رقم الهاتف .

```
PROGRAM  EXAMPEL3(INPUT,OUTPUT,FILES) ;
TYPE COSTUMERFILE = FILE OF TEXT ;
VAR
        NEW : COSTUMERFILE ;
        NUMBER,NAME,TEL : STRING[20] ;
        N : INTEGER ;
BEGIN
        ASSIGN (NEW,'A: TEST . TXT ') ;
        REWRITE (NEW) ;
        N:=5 ;
        WHIL N>0 DO
            BEGIN
                WRITE ('Enter-1 If You Want End') ;
                READLN (N) ;
                WHIL NOT EOLN  DO
                    BEGIN
```

```
                    WRITE ('Customer Number=');
                    READLN (NUMBER);
                    WRITE ('Customer Name =');
                    READLN (NAME);
                    WRITE ('Telephon=');
                    READLN (TEL);
                    WRITE('Enter-1If you want End
                           the Record');
                    READLN (N);
          WRITE(NEW,NUMBER,NAME,TEL);
          END;
          END;
     CLOSE (NEW) ;
END.
```

تمارين واسئلة عامة

1- إذا علمت أن :

TYPE Day Of Week =(SAT,SUN,MON,TUE,WED,THR, FRI);

فأوجد قيمة كل من :

1. PRED (MON)
2. ORD (SAT)
3. SUSS (TUE)
4. ORD (SUCC(THR))
5. ORD (PRED (MON))

2- اكتب برنامج بلغة باسكال لطباعة أسماء الأشهر الستة ورقم الشهر بالتسلسل إزاء كل واحد منهم مستخدما بيانات المستخدم التعريفية .

3- اكتب برنامج لإيجاد اكبر عنصر في مصفوفة ما مع دليله مستخدما بذلك الإجراءات .

4- اكتب برنامج لإيجاد اكبر واصغر عدد من مجموعة أعداد معطاه باستخدام الإجراءات .

5- اكتب برنامج بلغة باسكال لحساب لوغاريتم أي عدد باستخدام الاقترانات.

6- اكتب برنامج بلغة باسكال لإيجاد اكبر عنصر واصغر عنصر من عناصر مصفوفة A .

7- اكتب برنامج لإيجاد اكبر عنصر في كل سطر من عناصر مصفوفة ما .

8- اكتب برنامج لإيجاد اصغر عنصر في كل عامود من أعمدة مصفوفة ما .

9- اكتب برنامج لحل جملة معادلات باستخدام طريقة غوص جوردان .

10- اكتب برنامج ترتيب مجموعة من القيم المدخلة بشكل عشوائي ترتيبا تصاعديا.

11- اكتب برنامج لحساب درجات الطلاب المتقدمين لامتحان المقدمة ومقدار انحراف كل علامة عن المعدل ، مع العلم ان

$$\text{المعدل} = \frac{\text{مجموع علامات الطلبة}}{\text{عدد الطلبة}}$$

انحراف كل علامة عن المعدل = علامة الطالب المعدل .

على ان تبدو شاشة المخرجات بالشكل التالي :

```
AVERAGE = . . . . . . . .
  العلامة              الانحراف
 SCORE            DEVIATION
 ---------         ------------
 ---------         ------------
       *************
```

12- اكتب برنامج بلغة باسكال يتضمن ملف إدخال بيانـــات علـــى شـــكل سجلات حيث يتألف السجل من رقم الشـــخص ، اســـم الشـــخص ، الشارع ، المدينة ، رقم الهاتف وينتهي إدخال السجلات عند إعطاء القيمة صفر .

الفصل الحادي عشر

أساسيات بيئة باسكال

(11-1) الدخول إلى بيئة باسكال PASCAL والخروج منها :

بعد تشغيل جهاز الحاسوب وظهور محث النظام في الزاوية العليا اليسرى مـــن الشاشة كما يلي :   C:\>

ندخل الحرفين TP وهما اختصار لكلمتي Turbo Pascal ثم نضغط مفتاح الادخال ، أي

C:\>TP ↵

تظهر رسالة على الشاشة

Turbo Pascal version 7.0 copyright (c) 1983,92 Borland International .

وبعد هذه الرسالة تظهر نافذة الباسكال التي سنتوقف عند بعض تفاصيلـها في الفقرة التالية .

للخروج من بيئة الباسكال والعودة إلى شاشة الدوس يمكن اســتخدام عــدة أساليب إلا أن الطريقة السريعة للخروج من بيئة الباسكال أينما كنّـا فيــها هــو الضغط على مفتاحي ALT + X .

ملاحظـــة : يستخدم أحيانا للدخول إلى بيئة الباسكال كلمة TURBO أو TURBO PASCAL حسب اسم الملف التنفيـــذي المثبــت علــى جــهاز الحاسوب.

(2-11) أقسام النافذة الرئيسية لبيئة الباسـكال (نـافذة التنقيـح Edition Window)

النافذة الرئيسية أو نافذة التنقيح هي المكان الذي نكتـب وننقـح فيـه البرنامج المكتوب بلغة الباسكال بالاستعانة بنوافذ اخرى مسـاعدة ، ليصـار إلى تصميمه وتحويله إلى برنامج منطقي وتشغيله وتوثيقه بالاضافة إلى خدمات اخرى .
أجزاء هذه النافذة من الاعلى إلى الاسفل كما هو موضح في الشكل (1) :

1- شريط القوائم أو Menu Bar

ويضم عشرة اسماء ، تشكل عناوين للقوائم الفرعيه ، حيث كـل قائمـة تضم مجموعة من الأوامر التي تساعد المبرمج الوصول إلى المهام التي يريد انجازهـا بطرق مختصرة وسهلة ، هذه القوائم هي :

File – Edit – Search – Run – Compile – Debug – Tools – Options – Window – help .

عند تشغيل TP 7 (تربو باسكال 7) نلاحظ ان كل اسم قائمة يحتوي على حرف واحد بلون مختلف .

2- شريط العنوان Titele Bare ويضم من اليسار إلى اليمين :

أ. علبة الاقفال [ ] Close Box :

وظيفتها تمكين المستخدم من إقفال النافذة التي يعمل عليها بسرعة باستخدام الفارة . وذلك بوضع مؤشر الفارة على العلبة ثم النقر على الـزر الايسـر وبالتالي يتم اقفال النافذة الحالية وعرض النافذة التالية (أي الملف التالي) .

ب. عنوان النافذة Title :

ويقع في وسط الشريط يحتوي على عنوان النافذة التي تعمل عليها ، أي اسم الملف الذي يحتوي البرنامج الموجود في النافذة .

عند البدء بتشغيل TP 7 فـان نـافذة العنـوان تـاخذ الاسـم التـالي NONAME00.PAS أي بلا عنوان ، ولعنوان النافذة فائدتين :

1) تعريف الملف مع الامتداد الموجود على النافذة .

2) اذا كان قد فتحنا اكثر من نافذة في وقت واحد على الشاشة فيمكن الوصول إلى احدى النوافذ ووضعها في المقدمة ، وذلك بوضع مؤشر الفأرة على العنوان المراد تفعيله ثم ننقر بالمفتاح الايسر نقرة مزدوجة .

ج. رقم النافذة "_1_"

هذا الرقم يزداد مع زيادة عدد النوافذ التي نفتحها ضمن نافذة الباسكال ، ويساعد على تفعيل النافذة التي نريدها وذلك عن طريق مفتاح ALT ورقم النافذة المطلوبة .

والبديل عن ذلك يمكن استخدام القائمة Window والأمر Next أو القائمة Window و الأمر List ويظهر عندئذ صندوق حوار يعرض أسماء جميع الملفات المفتوحة على نافذة الباسكال نحدد احداها ونضغط مفتاح الادخال Enter .

```
 File  Edit  Search  Run  Compile  Debug  Tools  Options  Window  Help
[■]------------------------------ PRNFLTR.PAS ------------------------1-[]
end;

procedure ProcessCommandLine;
var
  Param: String;
  I: Integer;

  function ParamVal(var P: String; Default: Word): Word;
  var
    N, E: Integer;
  begin
    Delete(P, 1, 1);
    Val(P, N, E);
    if E = 0 then
      ParamVal := N
    else
      ParamVal := Default;
  end;

begin
  PrinterCodes := @AsciiCodes;
 === 206:1 ===
F1 Help  F2 Save  F3 Open  Alt+F9 Compile  F9 Make  Alt+F10 Local menu
```

الشكل (1) نافذة باسكال

د. علبة التحكم بحجم النافذة [↕] Zoom Box

نستخدم لتكبير حجم النافذة لتملا الشاشة بكاملها أو إعادة حجم النافذة إلى وضعها الطبيعي وذلك حسب شكل ظهور العلبة :

* اذا كانت من الشكل [↑] هذا يعني انه يمكن تكبير حجم النافذة اكبر لتملا الشاشة باكملها وذلك بالنقر على العلبة باستخدام الفارة .

* اذا كانت من الشكل [↕] يعني ان النافذة الفعالة في أقصى حجم لها ويمكن العودة إلى الحجم الطبيعي بالنقر على العلبة باستخدام الفارة .

والبديل عن استخدام الفارة هو القائمة Window والامر Zoom .

3- شريط التدرج العمودي Vertical Scroll

الموجودة في الجهة اليمنى من الشاشة ، ويعتبر شريط التدرج العمودي :

1. اداة لعرض الاجزاء المخفية من النافذة الفعالة باستخدام الفارة وذلك بالنقر على الاسهم ▼ أو ▲ .

البديل عن الفأرة استخدام مفاتيح الاسهم ↓ أو ↑ من لوحة المفاتيح .

2. مؤشر لتحديد الموقع المعروض بالنسبة للملف كاملا ، فإذا كانت علبة التدرج ☐ في منتصف شريط التدرج العمودي يعني أننا في منتصف الملف المفتوح على الشاشة .

ويمكن تغيير موقع علبة التدرج لعرض أجزاء أخرى من الملف وذلك باستخدام الفارة بالنقر على العلبة و السحب .

4- شريط التدرج الافقي Horizontal Scroll Bar

الموجود في اسفل الشاشة قبل الشريط الاخير ، ويعتبر هذا الشريط اداة لعرض الاجزاء المخفية من النافذة اذا كان الملف المعروض اكبر من استيعاب النافذة أفقيا وذلك اما بالنقر على الاسهم ◄ أو ► التي تتحرك باتجاه

اليمين أو اليسار . واما باستخدام الاسـهم ←أو → مـن لوحـة المفاتيح.

ملاحظة : اذا نقرنا بالفارة على المساحة المظللة الواقعة بين الاسهم في شـريط التدرج ننتقل صفحة كاملة في الملف (أي نافذة كاملة Window Ful).

5- موقع الاشارة "1.1" Cursor Location

هذا الموقع يبين احداثيات المشيرة Cursor (الوامض) على النـافذة وتدعـى المشيرة باسم مؤشر الادراج .

مثلا اذا كانت من الشكل 1.1 يعني ان المشيرة في السطر الأول والعمـود الأول من النافذة أي انها في الزاوية العليا اليسرى

اذا كان من الشكل 3.5 يعني ان المشيرة في السطر الثالث والعمود الخامس .

6 – النافذة الفعالة Active Window

وهي المنطقة الواقعة بين شريط العنوان وشريط التدرج الافقي والعمـودي والتي يمكننا ادخال أوامر الباسكال وتنقيحها وتشغيل البرنامج باسـتخدام القوائم

- وتستوعب النافذة إلى 256 حرفا (رمزا) على الاكثر في الخـط الافقـي الواحد .
- مهما فتحنا نوافذ على الشاشة فإنه هناك نافذة واحدة فقط فعالة في وقـت واحد .
- أي امر نختاره أو نص يطبق على النافذة الفعالة .

7 – شريط الرسائل Message Bar

يعطي هذا الشريط رسالة مختصرة جدا عن المهمة التي نريد القيام بهـا بعـد انجاز الخطوات المناسبة بها .

عند بداية تشغيل TP 7 تحوي الرسالة المعلومات التالية :

F1 Help F2 Save F3 Open Alt-F9 Compil F9 Make F10 Menu .

الا ان هذه الرسالة غير ثابتة فهي تتغير مع تغيير المهمة التي ننوي انجازهــا ، الا ان كل ما يستطيع TP 7 القيام به في هذه اللحظة من مــــهمات هــو الانجازات الستة 6 المعروضة في الرسالة ومنها نبدأ بإنجاز مــهمات اخــرى تدريجيا .

(3-11) تشغيل شريط القوائم :

لتشغيل شريط القوائم ، وفتح القوائم المتفرعة منه نتبع احدى الطرق التالية :

1) نضغط على المفتاح F10 فيصبح شريط القوائم فعال ثم نســتخدم مفــاتيح الاسهم ← أو→ لتحديد اسم القائمة التي نريد فتحــها ثم نضغــط مفتاح الادخال Enter .

2) نضغط المفتاح Alt مع الحرف المميز من اسم القائمة المراد فتحها ، مثلا عنــد الضغط على مفتاح Alt مع الحرف F نفتح القائمة File.

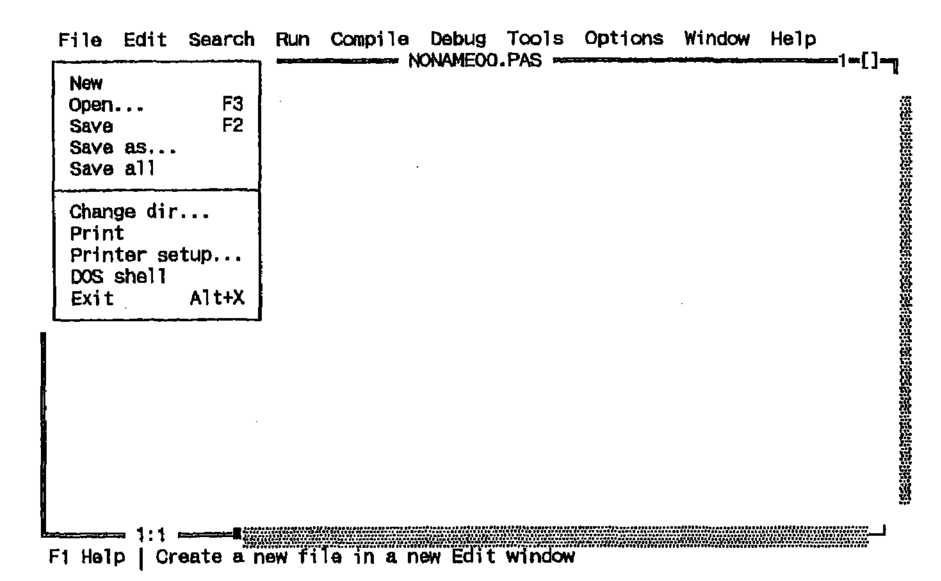
3) باستخدام الفارة ، نضع مؤشر الفارة على اسم القائمة المراد فتحــها وننقــر بالزر الايسر فتنفتح القائمة .

بعد فتح القائمة ، بغض النظر عن الطريقة المتبعة، نلاحظ ان كــل قائمــة تحوي على مجموعة من الأوامر الشكل (2). ولتشغيل أحد الأوامر نتبــع إحــدى الطرق الثلاث التالية :

أ. استخدام مفاتيح الأسهم ↓ أو ↑ لتحديد الأمر ثم نضغط مفتــاح الإدخال Enter .

ب. الضغط على الحرف المميز من اسم الأمر .

ج . باستخدام الفارة بوضع مؤشرها على الأمر المطلوب والنقر على الــزر الايسر .



الشكل (2) أوامر احد القوائم

وقبل ان نتعرف على وظيفة كل امر من أوامر TP 7 لا بد من الاشـارة إلى انه هناك مواصفات عامة لاسلوب العمل في بيئة الباسكال :

1) وجود ثلاثة نقاط امام الامر : هذا يعني ان الأمر يحتـاج إلى تفصيـلات معمقة اكثر لتنفيذه وبالتالي عند اختيار الأمر يظهر على الشاشة صنـدوق يدعى صناديق الحوار (Dialog Boxe) نستطيع تحديد بعض الخيـارات اللازمة ، وسنتناول صناديق الحوار في فقرة مستقلة .

2) وجود العلامة ► جانب الأمر ، هذا يعني ان هنالك قائمة فرعيـة تظـهر عند تنفيذ هذا الأمر .

3) هناك بعض الأوامر لونها باهت أو يختلف لون أحرفها عن لون باقي الأوامر: هذا يعني إن الأمر متوقف عن العمل مؤقتا ريثما تتوفـر بعـض الشـروط الملائمة ليأخذ لونه الطبيعي وبالتالي امكانية تنفيذه ، مثل الامر Cut قبـل تحديد النص المختار لقصه .

4) وجود اسماء مفاتيح امام بعض الأوامر ، هذه المفاتيح هي عبارة عن اسلوب مختصر لتشغيل الامر من غير الرجوع إلى القائمة .

(4–11) صناديق الحوار Dialog Boxes

بشكل عام صندوق الحوار يأخذ شكل النافذة ويحتوي على علبة الاقفال[.] Close Bar التي تساعد على اقفالها (إلغائها) في حالة استخدام الفأرة . ولا تحتوي على شرائط التدرج وعلبة تغيير الحجم ، ويضم أي صندوق حوار بعض أو كل العناصر التالية :

أ) علب الاختيار [ ] Check Boxes

تعطي امكانية اختيار اكثر من علبة في نفس الوقت ، وعند اختيار العلبة التي نريدها يتم وضع اشارة X داخل القوسين [ ] .

ب) أزرار الخيارات ( ) Radio Buttons

لا يمكن اختيار اكثر من زر ضمن الجدول الواحد ، وعند اختيار احد الازرار التي تريدها يتم وضع نقطة في وسط القوسين ( ) .

ج) أزرار الضغط Push Butions

تستخدم لابلاغ TP 7 ماذا سيفعل بالخيارات والقرارات التي حددناها من صندوق الحوار وعادة توجه ثلاثة ازرار :

OK لتشغيل الخيارات والقرارات المتخذة

Cansel إلغاء العملية كلها .

Help طلب مساعدة وتوضيحات حول علبة المحادثة .

د) جداول الاختيار ↓ List Options

عندما نريد إدخال بيانات ونحتاج إلى مساحة اكبر من المساحة المخصصة في علبة الحوار ، أو أننا نريد أختيار اشياء مخفية من المساحة المخصصة في صندوق الحوار، نلاحظ وجود السهم↓ جانب المساحة المحدودة وظيفته

إظهار المخفي من المساحة المخصصة أو توسيع المساحة لامكانية الادخال .

كيفية التنقل داخل صندوق الحوار :

1. نستخدم مفتاح TAB أو Shift + TAB للانتقال من زر ضغط إلى آخر .
2. نستخدم مفاتيح الاسهم للانتقال من علبة اختيار إلى أخرى ومن زر خيار إلى آخر .
3. نستخدم المفتاح TAB للانتقال من جدول أو قسم إلى آخر في صندوق الحوار.

(11-5) بعض المهمات التي يمكن القيام بها :

(11-5-1) فتح ملف جديد :

نختار الامر New من قائمة File ، عندئذ تظهر نافذة تنقيح جديدة بالاسم الافتراضي : NONAME00.PAS ، وتظهر المشيرة Cursor في الزاوية العليا اليسرى ، يمكننا البدء بكتابة تعليمات وأوامر البرنامج المراد ادخالها عند المشيرة .

(11-5-2) فتح ملف مخزن مسبقا :

نختار الامر Open من قائمة File ، عندئذ يظهر صندوق حوار باسم Open a File انظر الشكل (3) والذي يتألف من :

1 - حقل Name لإدخال اسم الملف المراد فتحه .

2 - حقل Files يضم جميع الملفات ، والفهارس المتوفرة على الديسك الصلب Hard Disk ، يمكننا بالنقر المزدوج على الفهرس الذي يحوي ملفات باسكال عندئذ تظهر جميع الملفات ، وبالنقر على احد الملفات المراد فتحها ينتقل الاسم إلى حقل Name ثم نستخدم الزر Open لفتحه (أو يمكن بالنقر المزدوج على اسم الملف) .

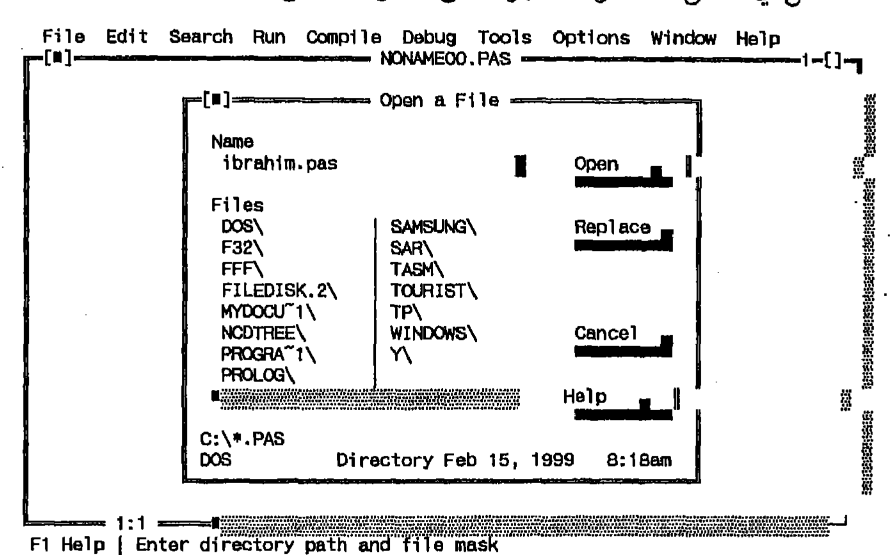
3 - الأزرار Open لفتح الملف المختار .

Replace لاستبدال الملف المختار بملف آخر .

Cancel إلغاء صندوق الحوار Open وجميع التعديلات التي تمت داخله

Help إعطاء بعض المعلومات عن كيفية التعامل مع صندوق الحوار

4 - حقل في اسفل الصندوق يظهر بعض المعلومات عن الملف المختار .



الشكل (3) صندوق حوار فتح (**Open File**)

(3-5-11) تخزين ملف باسم جديد

نختار الامر Save As من قائمة File ، عندئذ يظهر صندوق حوار باسم Save File As ، انظر الشكل (4) ، والذي يحوي على :

1. حقل Save File As : لإدخال الاسم الجديد للملف .

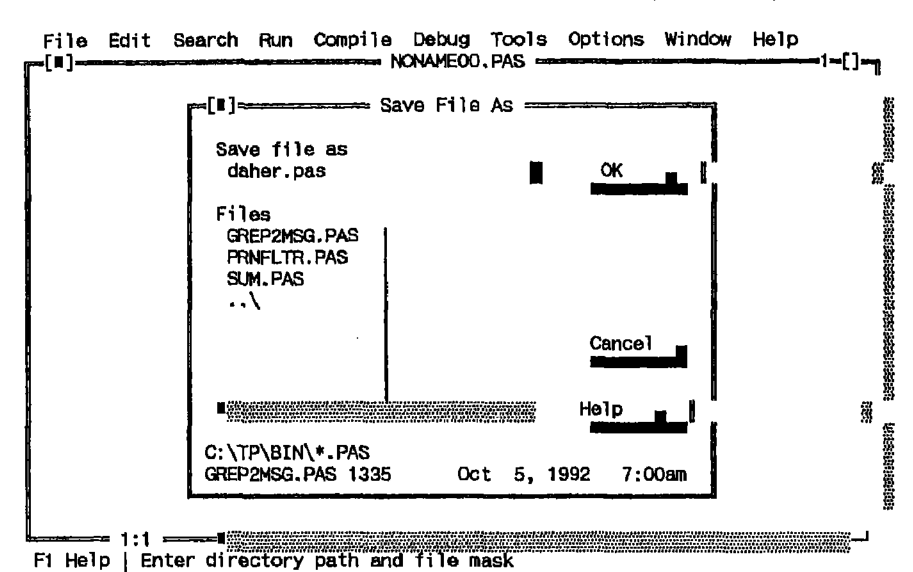
2. حقل Files : يضم الفهارس المتوفرة ، فإذا أردنا تخزين الملف على فـــهرس معين أو في سواقة معينة فعندئذ ننقر اسم الفهارس أو السواقة.

3. الأزرار OK: لإتمام عملية التخزين .

Cancel لإلغاء صندوق الحـــــوار حقـــل Save File As وجميـــع التعديلات التي تمت داخله .

Help إعطاء بعض المعلومات عن الملف المراد تخزينه .

4. حقل في الاسفل يظهر بعض المعلومات عن الملف المراد تخزينه .



الشكل (4) صندوق حوار تخزين ملف Save As

(11-5-4) تخزين ملف بإسم قديم

نختار الامر Save من قائمة File ، عندئذ يقوم الحاسوب بتخزين الملـــف مع جميع التعديلات التي تمت عليه تحت الأسم القديم الموجود في شريط العنوان .

(11-5-5) طباعة ملف

نختار الأمر Print من القائمة File ، فيتم طباعة الملف المفتوح الموجـــود داخل النافذة الفعالة ، أما اذا أردنا طباعة مقطع معين من النا فذة ، عندئذ نضغـــط المفتاحين Ctrl + K ثم المفتاح P .

(6-5-11)استخدام ميزة النسخ والقص واللصق والمسح عند كتابة البرنامج

لاستخدام هذه الميزات يجب اتباع ما يلي :

1- تحديد المقطع المراد تطبيق عليه ميزة النسخ أو القص أو المسح ويتـــم ذلـــك باحدى الطريقتين :

أ) باستخدام الفأرة : نضع مؤشر الفأرة في بداية المقطع المراد تحديـــده ثم نضغط الزر الأيسر مع السحب على المقطع المراد تحديده .

ب) باستخدام لوحة المفاتيح : نضع المشيرة (إشارة الأدراج) على بدايـــة المقطع ثم نضغط مفتاح shift مع أحد مفاتيح الأســـــهم بالاتجـــاه المناسب .

2- بعد تحديد المقطع يمكننا من استخدام أحد الأوامر التالية من القائمة Edit

Cut حذف المقطع المحدد ووضعه في الحافظة clipboard .

Copy وضع نسخة من المقطع المحدد في الحافظة clipboard .

Clear حذف المقطع المحدد نهائيا ، دون وضعه في الحافظة clipboard

3- لاستخدام المقطع المنقول إلى الحافظة بواسطة الأمر cut أو copy ، نقـــوم بوضع المشيرة (مؤشر الأدراج) في المكان المراد وضع المقطع ثم نختار الأمــــر paste فيتم نسخ المقطع الموجود في الحافظة .

بالإضافة الى ذلك هناك بعض المفاتيح في لوحة المفـــــاتيح Keyboard تســـاعد المستخدم على العمل داخل نافذة باسكال ، وهذه المفاتيح سنعرضها في الجـــــدول التالي :

وظيفته	اسم المفتاح
الانتقال حرف واحد في كل مرة إلى اليمين	
الانتقال حرف واحد في كل مرة إلى اليسار	
الانتقال كلمة واحدة في كل مرة إلى اليمين	Ctrl +
الانتقال كلمة واحدة في كل مرة إلى اليسار	Ctrl +
نقل المشيرة إلى بداية السطر الحالي	HOME
نقل المشيرة إلى نهاية السطر الحالي	END
نقل المشيرة إلى أول سطر في النافذة	Ctrl+Home
نقل المشيرة إلى أخر سطر في النافذة	Ctrl+END
نقل المشيرة بمقدار النافذة المفتوحة إلى الأعلى	Pg Up
نقل المشيرة إلى بمقدار النافذة المفتوحة إلى الأدنى	Pg Dn
نقل المشيرة إلى بداية الملف	Ctrl+Pg Dn
نقل المشيرة إلى نهاية الملف	Ctrl+Pg Dn
حذف الحرف الواقع فوق المشيرة ويستخدم لحذف سطر فارغ بعد نقل المشيرة إلى بداية السطر الفارغ .	Del
حذف الحرف الواقع على يسار المشيرة	Backspace
طباعة الحرف الكبير	احد الازرار+shift
نقل المشيرة مسافة على الأغلب بمقدار 4 أحرف	Tab

(11-5-7) تصحيح الملف قواعديا وترجمته إلى لغة الآلة:

نستخدم الأمر compile من القائمة ، عندئذ يقوم الحاسوب بتدقيق صحة كتابة أوامر وتعليمات باسكال ، فإذا وجد خطأ في مكان ما في البرنامج عندئذ يعرض رسالة في أعلى النافذة برقم الخطأ ونوعه ، ويضع المشيرة على السطر الموجود فيه الخطأ .

(11-5-8) تشغيل البرنامج

نختار الأمر RUN من القائمة RUN ، عندئذ يبدأ الحاسوب بتشغيل البرنامج المفتوح على نافذة باسكال ويعطي النتائج على شاشة مستقلة تدعى شاشة النتائج .

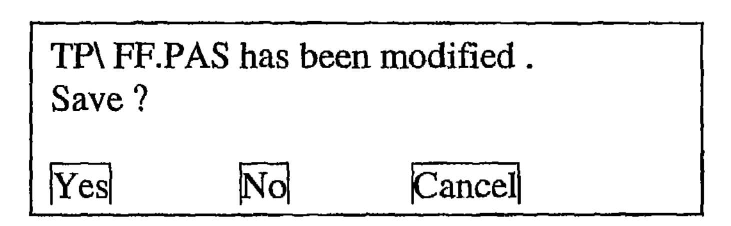
(11-5-9) عرض نتائج تشغيل البرنامج

نختار الأمر user screen من قائمة Debug (وفي بعض إصدارات الباسكال يكون الأمر موجود في قائمة window) .

(11-5-10) الخروج من بيئة باسكال :

نستخدم الأمر Exit من القائمة file .

اذا كان الملف أو الملفات المفتوحة غير مخزنة أو أجري تعديلات عليها دون تخزين عندئذ تظهر رسالة :


TP\ FF.PAS has been modified .
Save ?

Yes No Cancel

هذه الرسالة تشير إلى أن الملف TP\FF.PAS يحتاج إلى تخزين ، فاذا أردنا تخزينه وتخزين التعديلات نختار الأمر Yes واذا كنا لا نريد تخزينه وتخزين التعديلات نضغط زر No ، واذا رغبنا بعدم الخروج من الباسكال نضغط cancel

المراجع العربية

1. ريان (أحمد)-خدمات الانترنت-منشورات المجمع الثقافي-أبو ظبي 1997
2. دوفور(ارنورد) انترنت-ترجمة- الدار العربية للعلوم-1998 .
3. نشاوي (محمد أسعد) تعلم ويندوز 95 باعتماد اللغة العربيـة-دار شـعاع للنشر والعلوم-حلب 1997.
4. رحو (غازي ابراهيم)- الصباغ (عماد)- علم الحاسوب والبرمجـة لطلبـة الجامعات ، دار زهران للنشر والتوزيع 1997 .
5. برهان (محمد نور) رحو (غازي ابراهيم) نظم المعلومـات المحوسـبة-دار المناهج-1998 .
6. العبدالله (سعد الدين) الخوارزميات والبرمجة-منشورات جامعـة حلـب-1997.
7. الصفدي (محمد سالم) نظم معالجة البيانات والبرمجة بلغة باسكال-دار وائـل للنشر-1999.
8. الشرايعة (احمد عبد العزيز) واخرون الحاسوب والبرمجيات الجـاهزة-دار وائل للنشر-1998.
9. نائب (ابراهيم عبدالواحد) رحو (غازي ابراهيم) - المرجع لمختبرات علـم الحاسوب-دار المناهج 1998.
10. مجلة الكمبيوتر والاتصالات-سلسلة الاعداد لعام 97-98.
11. مجلة PC MAGAZINE ، سلسلة الاعداد للعام 97-98-99 .
12. دليل المستخدم Micro Soft Windows 3.1 شركة ميكروسـوفت- 1994 .
13. دليل المستخدم Micro Soft Excel 5 شركة ميكروسوفت-1993.

14. الفيومي (محمد احمد) مقدمة في علم الحاسوب والبرمجة بلغة باسكال - دار الفلاح 1997 .

15. أكرم (كلير) - البرمجة بلغة الباسكال - دار ماكروهيل ، الدار الدولية للنشر والتوزيع 1988 .

16. منصور (عوض) - أبو النور (محمد) - مقدمة في علم الحاسب الالكتروني وبرمجة بيسك الأردن 1987 .

17. الراغب (فوزي) نظرة إلى أعماق الكمبيوتر منشورات المركز الحديث للكمبيوتر 1985 .

18. الرماحي (سامي) الحاسوب (شرح تعليمي مبسط) الدار العربية للعلوم 1988 .

19. طلبة (محمد فهمي) الحاسبات الالكترونية (حاضرها ومستقبلها) موسوعة دلتا كمبيوتر 1992 .

20. رحو (غازي ابراهيم) الفيومي (محمد أحمد) الحاسوب وزارة التربية العراقية العراق 1994 .

المراجع الاجنبية

1. Addyman , A, M. Et A (1979) , Adraft Description Of Pascal , Pascal New 14 And Software , Practice And Experience .
2. Atkinson , L. V. (1980)-Pascal Programming John Wiley And Sons Limited .
3. Atkinson , L. V. (1982)-A Students Guide To Programming In Pascal John Wiley And Sons Limited .
4. Barron D. W. (1981)-Pascal : The Language And Its Implementation John Wiley And Sons Limited .
5. Pindly W. And Watt Limited . Pascal An Introduction To Methodical Programming Pitman Publishing Ltd.
6. Jenson, I And Wirths, N. (1975) Pascal-User Manual And Report (Springier-Verlag, New Yourk) .
7. Matnszek , D. L. - Quik Pascal . John Wiley And Sons Limited .
8. Moore (1980)- Foundation Of Programming With Pascal . John Wiley And Sons Limited .
9. Pemertcn S. And Danieis M. C.- Pascal Implementation .(1983)
10. Jones , R. M. (1983) An Implementation Of Data Structure Allyn And Bacon Inc.
11. Schn Eider G. M. And Bruell Sec (1981) Advanced Programming And Problem Solving With Pascal.
12. Weish , J , Sneeringer , W. J. And Hoare , C., A.(1977)

Ambiguities And Insecurities In Pascal, Software Practice And Experience .

13. Wirth, N. (1975a)
An Assessment Of The Programming Language Pascal Sigplan Notices,

14. Wirth, N, (1975b) - Algorithms + Data Structure = Programs
(Prentice-Hall Englewood Cliffe , N. J.)

15. Wiison I. R. And Addsyman A. M. (1978)
A Practical Introduction To Pascal .
The Macmillan Presc LTD.

16. Michael Shave - Data Structure
McGrow Hill (1975)

17. Bertziss, A. T. (1975) - Data Structure Theory And Practices .
Academic Press, London.

18. Elson , M. (1973) - Concepts Of Programming Languages
Sci. Research Ass, London.

19. Knuth, D. G. (1973)
The Art Of Computer Programming VOL1, VOL2, VOL3 Addisam – Wealeg London .

20. Meek, B. And Heath , D. (Ed) (1980)
Guide To Good Programming Practice .
Ellis Horwood.

21. John Beidler (1982) - An Introduction to Data Structures
Allyn And Bacon

22. Dijkstra , E. W. (1976) - A Principle Of Programming
Prentice – Hall

23. Turbo Pascal – Elliot B. Koffman With Bruce Maxim
4th Edition – Addison – Wesley Publishing Company (1996)

24. Weizenbaum, Joseph, Computer Power And Human Reason W. H. Freeman, San Francisco (1976)

25. Kernighan Brian W. And Dennice M. Ritchie The Programming Language Prentice-Hall, Englewood Cliffs NJ, (1978) .
26. Postel J. B. , Editor : Internet Official Protocol Standards RFC-1880 , STD-1, Internet Architecture Board,1995
27. Mitzi Walts , The Internet International Directory Zift Davis Press-1995
28. Colman P. And Dyson P. - Mastering Internet, The Windows 95/NT Edition , SYBEX 1997
29. Metzelar F.Scharp F.Scharp F.- Projects For Dos And Windows 3.1 , Benjamin / Comings – 1995 .
30. Larry Long – Nancy Long – Introduction To Computers And Information Systems – 5^{th} ,Ed . , Printice Hall , Inc .
UK ,1997 .
31. Slothink D.L.And Others – Computers And Applications- Second Edition , D.C. Meath And Company 1994 .
32. Marija Jnorusis – SPSS For Windows (Base System User's Gulde , Release 6.0)- Printed In The United Ststes Of America 1993 .
33. Crumlish Christian – The ABCS Of The Internet – SYBXX Inc., 1996 .

الإختبارات

الهدف :

تقديم بعض التمارين غير المحلولة وعلى الطالب حلها داخـــــل قاعة المحاضرة كاختبار لإظهار مقدرة الطالب في بعض مواضيع هذا الكتاب .

الاختبار (1)

1- حول العدد التالي :$(57.125)_{10}$ = إلى مكافئـــه في النظـــام الثنـــائي والثماني والسادس عشري ؟

2- حول الأعداد التالية إلى النظام الثنائي :

$(37.6)_8 =$
$(59ED)_{16} =$
$(462.7)_{10} =$

الاختبار (2)

1- أوجد ناتج العمليات التالية :

$(101101)_2 + (11101.11)_2 =$

$(573.42)_8 + (22.41)_8 =$

$(ED4.F)_{16} + (AA)_{16} =$

$(11001)_2 - (111)_2 =$

$(765)_8 - (32)_8 =$

$(B32)_{16} - (32)_{16} =$

الاختبار (3)

1- أوجد المتمم الحسابي الثنائي الأول والمتمم الحسابي الثنائي الثاني لكـــــل مـــن الأعداد :

1- $(-53)_{10}$

2- $(36)_{10}$

باستخدام حاسوب طول الكلمة فيه يساوي 8 ثنائيات ؟

3- أوجد ناتج مايلي باستخدام حاسوب طول كلمته يساوي 8 ثنائيان ؟

$(82)_{10}$ $\quad 60)_{10} =$

الاختبار (4)

1- أكتب خوارزمية وخريطة سير العمليات للمشكلة التالية :

يأخذ بائع عمولة على مجمل مبيعاته حسب ما يلي :

0.02 اذا كانت مجمل مبيعاته أقل من 500 دولار

0.05 اذا كانت مجمل مبيعاته من 500 دولار إلى 1500 دولار

0.15 اذا كانت مجمل مبيعاته أكثر من 1500 دولار

والمطلوب ايجاد مقدار العمولة بالعملة المحلية التي يأخذها البائع .

الاختبار (5)

1- أكتب خوارزمية وخريطة سير العمليات لايجـــاد مضـــروب العـــدد N حسب القاعدة التالية :

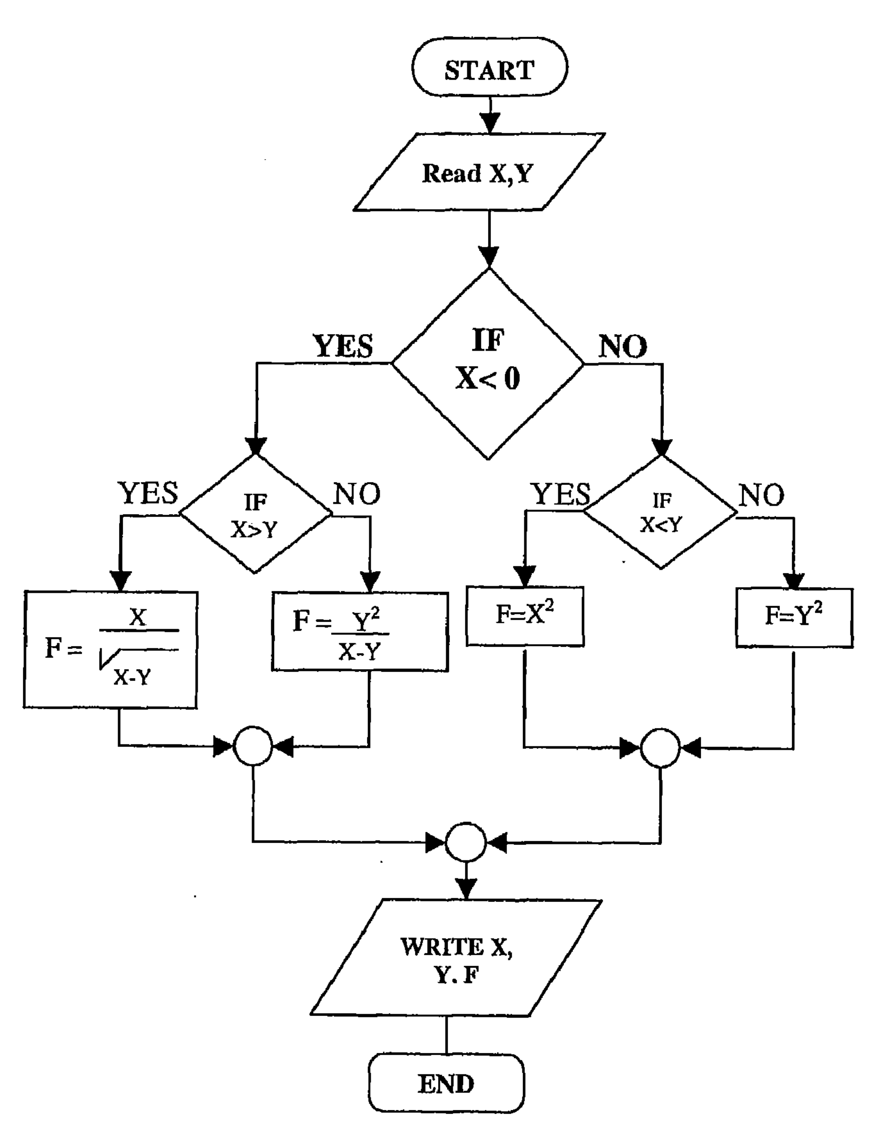
$$N! = \begin{cases} 1 & ; N = 0 \\ 1*2*3*......*N & ; N <> 0 \end{cases}$$

الاختبار (6)

أكتب برنامج بلغة باسكال لإيجاد عدد الطلاب الراسبين وعدد الطلاب النـاجحين وعدد الطلاب الذين تزيد علاماتهم عن 70، لطلبة شعبة المقدمة حيث أن عـــدد الطلبة المتقدمين للامتحان يساوي 60 طالب .

الاختبار (7)

1- المخطط التالي يمثل مخطط سير العمليات لمشكلة ما ، والمطلوب كتابة برنامج بلغة باسكال المقابل لهذا المخطط ، وبين ما هي النتائج على افتراض أن البيانات المدخلة $x = 5$, $y = 3$.



الاختبار (8)

1- هل كتابة جمل وأوامر هذا البرنامج صحيحة ، بين الأخطـــاء الموجــودة وصححها ؟

```
PROGRAM EXAM (INPUT,OUTPUT);
VAR
        X,Y : REAL
BEGEN
        X = 1.0;
        WRITELN ('X        Y');
        WRITELN ('---      ---');
        WHELE  X < = 6.0  DO
        BEGIN
                Y:=X*X*X ;
                WRITELN (X:5:2,'    ',5:2);
                X:=X+0.5;
        END;
END.
```

الاختبار (9)

1- بين نتائج تنفيذ كل من البرامج التالية :

```
1) PROGRAM  EX1 (Input,Output);
  VAR
            M     : REAL ;
            J,K,L : INTEGER ;
   BEGIN
            J:= 4;
            K := J+2 ;
            L := J – 5 ;
            M := J- 6 ;
   END.
```

النتائج

```
2) PROGRAM E2 (Input,Output);
   VAR
      I,J :INTEGER;
   BEGIN
      FOR  I:= 1 TO 5 DO
      BEGIN
            I:= I + 1;
            I:= I * I;
            WRITELN ('I=',I:4,'J=',J:4);
      END;
      WRITELN ('**********');
   END.
```

النتائج :

الاختبار (10)

1- بين نتائج تنفيذ البرنامج التالي :

```
1) PROGRAM  EX1(Input,Output);
   VAR
          X: ARRAY [1..5] OF REAL;
          I: INTEGER; SUM,A:REAL;
   BEGEN
          SUM :=  0.0;
          FOR I:= 1 TO 4 DO
          BEGEN
                 SUM :=SUM + X[I];
                 WRITELN (SUM,'      ',);
          END;
          A:= SUM/4;
          WRITELN ('SUM =' ,SUM,'A=',A);
   END.
```

النتائج :

الاختيار (11)

أكتب برنامج بلغة باسكال لضرب مصفوفة A أبعادها (5 X 4) بثابت K ؟

الاختبار (12)

البرنامج التالي بداية برنامج لإيجاد أكبر عنصر في مصفوفة A أبعادها (4,4) ، أكمل البرنامج بحيث يطبع اكبر عناصر المصفوفة المدخلة ؟

```
PROGARM   EXAMPLE (INPUT,OUTPUT);
VAR
       A: ARRAY [ 1..4] OF REAL ;
       I,J,MAX: INTEGER;
BEGIN
              WTITELN ('Enter Data of Matrix A :');
              FOR I:= 1 TO 4 DO
                     BEGIN
                     FOR J:= 1 TO 4 DO
                            READ ( A[I,J] );
                            WRITELN ;
                     END;
```